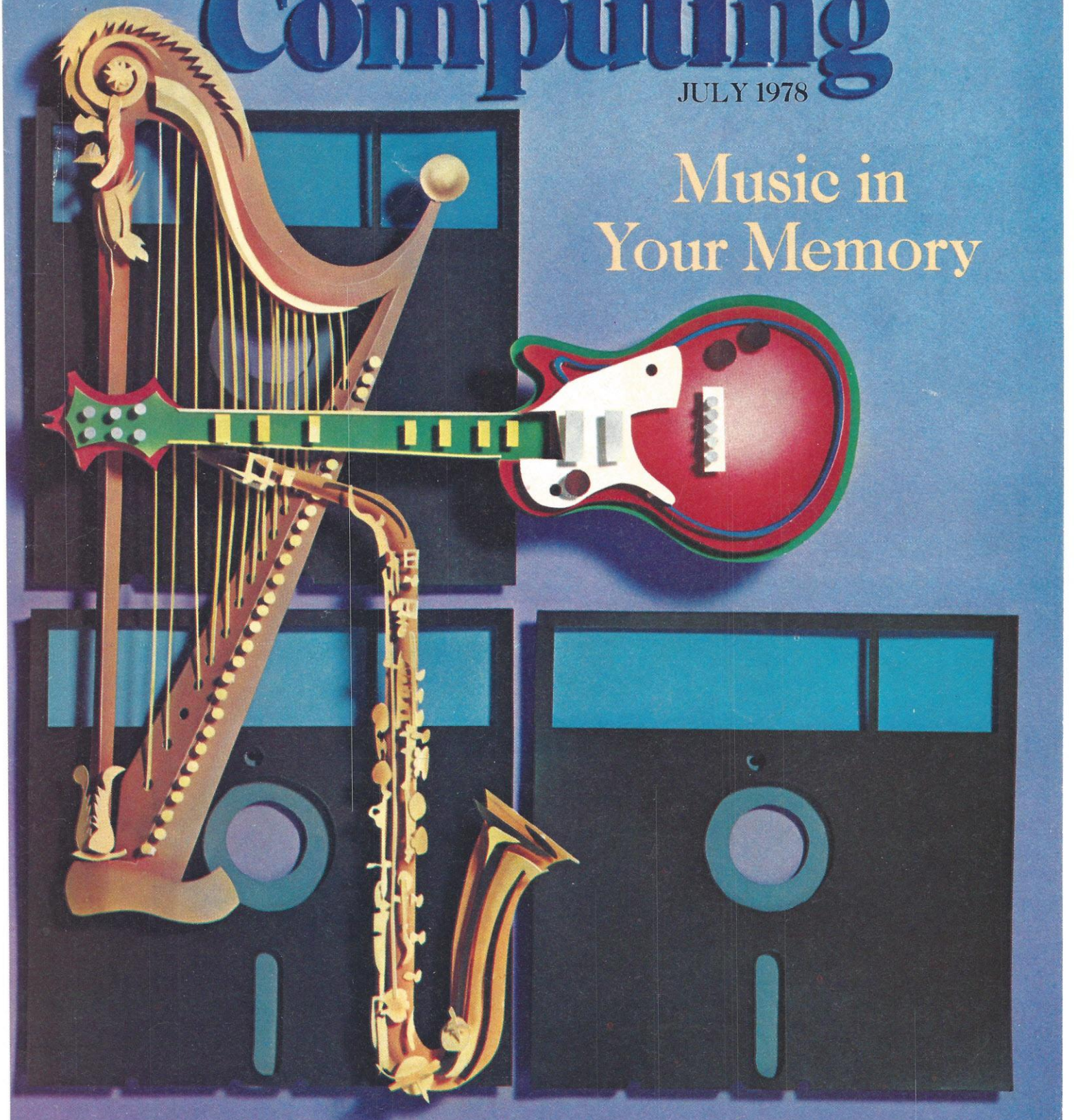
Small Business Systems · Batter-up · Terminals, CRTs & Keyboards

$2.00

Personal Computing

JULY 1978

Music in Your Memory



Radio Shack's personal computer system?
This ad just might make you a believer.

You can't beat the 4K system at $599

TRS-80 "Breakthru"
- TRS-80 microcomputer
- 12" video display
- Professional keyboard
- Power supply
- Cassette tape recorder
- 4K RAM, Level-I BASIC
- 232-page manual
- 2 game cassettes

...or the step-up 16K system at $899

TRS-80 "Sweet 16"
- Above, except includes 16K RAM

...or the fast 4K/printer system at $1198

TRS-80 "Educator"
- Above, except includes 4K RAM and screen printer

...or the Level-II 16K/printer/disk system at $2385

TRS-80 "Professional"
- Above, except includes 16K RAM, disk drive, expansion interface, and Level-II BASIC

So how are you gonna beat the system that does this much for this little? No way!

...The amazing new 32K/Level-II/2-disk/ line printer system at $3874

TRS-80 "Business"
- Above, except includes 32K RAM, line printer, and two disk drives

Get details and order now at Radio Shack stores and dealers in the USA, Canada, UK, Australia, Belgium, Holland, France, Japan. Write Radio Shack, Division of Tandy Corporation, Dept. C-045, 1400 One Tandy Center, Fort Worth, Texas 76102. Ask for Catalog TRS-80.

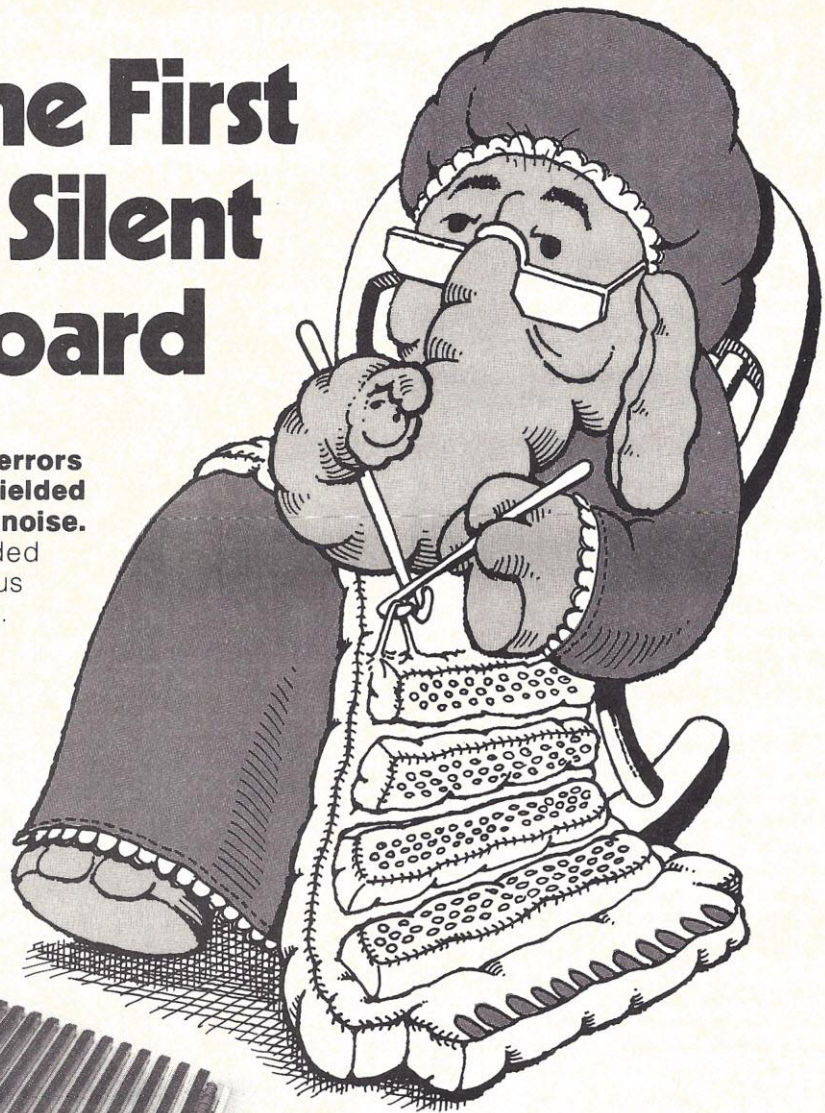
Radio Shack®

The biggest name in little computers

Prices May Vary at Individual Stores and Dealers

ARTEC CRAFTSMANSHIP HAS CREATED

The First Truly Silent Motherboard



Noise in your bus lines means errors in your programs. The Artec shielded Motherboard totally eliminates noise.

At 4MHz, the Artec shielded Motherboard is free from spurious noise. No ringing in your bus lines. No errors in your programs.

This Motherboard offers you engineering and craftsmanship never before available in the small computer field. Outstanding as either a replacement for your present Motherboard or as the heart of a new system. Consider these features:

- ⅛th inch thick—more than twice as thick as most Motherboards.
- Totally shielded—all holes plated through; full bus terminations.
- Fits easily into any standard chassis.
- Masterite edge connectors—the finest quality connectors available.
- Reflowed solder circuitry.
- No soldering required.
- Designed for the S-100 bus.

The Motherboard price is: $150(KIT)
$190 (ASSEMBLED)

® Trademark of Digital Equipment Corporation

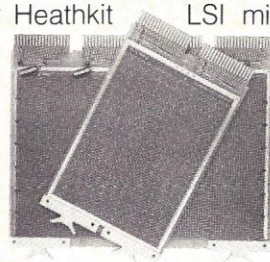
Five years of experience in every card

For five years, Artec has worked hard to develop a complete line of custom, prototype and off-the-shelf printed circuit boards. And in five years of tough industrial use, Artec boards have proven themselves among the most reliable boards available anywhere.

NEW! DEC® and Heath Compatible LSI Boards

The new Artec WW11 lets you adapt or add onto your DEC LSI-11 or Heathkit LSI minicomputer. Can accommodate 14 and 16 pin DIPs plus all necessary passive components.

FULL CARD $75
(10.45" x 8.4")
HALF CARD $35
(5.225" x 8.4")



Order today!

Put an Artec board to work for you. Use your Mastercharge or Visa. Or just send along a money order. We can accept only U.S. currency. Please include $3 handling on all orders. California residents add 6% sales tax.

Please send me: (include quantity)

___ Shielded Motherboard ___ Full WW11 Card ___ Half WW11 Card

☐ I've enclosed a money order for $ _____

☐ Mastercharge ☐ Visa _____ (number) _____ (exp. date)

Name _____

Address _____

City _____ State _____ Zip _____

Calif. Res. add 6% sales tax. Encl. $3 handling. 10% discount for students and computer club members. (Please enclose name of club or school)

ARTEC ELECTRONICS, INC.

Artec Electronics, Inc. • 605 Old County Rd. • San Carlos, CA 94070
(415) 592-2740

CIRCLE 4

AT THE COMPUTER FACTORY

JULY IS A VERY SPECIAL MONTH

Act before August 20th, and you can save between $150 and $600

Just look at these APPLE II specials.

VOICE CONTROL

The Apple II, known to date for its unique color graphics, has been party to the most far-reaching computer development of the late 70's . . . ON/OFF VOICE CONTROL.

HEURISTICS and MOUNTAIN HARDWARE, following many years of development, have just introduced low-cost voice-recognition and controller systems designed specifically for the Apple II.

The Speech Lab by HEURISTICS . . . complete with board, microphone and programs. Recalls 32 Spoken Words.
Your Apple II can react to your own voice command.

The Introl by MOUNTAIN HARDWARE . . . complete with board, demo tape, and manuals. Ability to control two AC outlets. No extensive wiring needed. Utilizing your own AC lines.

The Computer Factory has configured the Dream System Speech Lab and Introl, alongside your Apple II, gives you a taste of the future, now. Who knows what is next?

SPECIAL

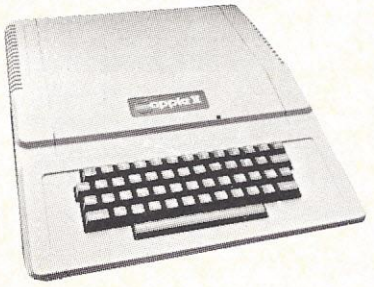
Apple II 16K	$1,195.00
Mountain Hardware	
Control Board	189.00
Introl	149.00
HEURISTICS	
Speech Lab	189.00
sugg. mfg. retail price	~~$1,1722.00~~
OUR PRICE	$1,572.00
	SAVE $150

PERSONAL

Our Special Personal System includes Apple II's 116 K DiskII. Disk allows quick storage and easy access to files and large quantities of data, for both home and business use. Speeding up load time and data retrieval, the Disk adds flexibility.
For home use, Expandors Impact Printer is undoubtedly the best value on the market today. Full 80 column printer . . . . 8½ inch paper . . . . . (NOT thermal)

SUPER SPECIAL

Apple II 32K	$1,495.00
Disk II (116K) mini-floppy	495.00
Printer Interface	218.75
Expandor Printer	425.95
sugg. mfg. retail price	~~$2,634.70~~
OUR PRICE	$2,395.00
	SAVE $239.30



BUSINESS

There is only one word to describe our business package. VALUE! 48k Apple II . . . . . Dual Mini Floppy Disks Drives, yielding approx. 232k bytes of storage. Centronics 779 Printer . . . . 80-132 columns . . . . 60-110 cps . . . . perfect for all business forms . . . . multicopy . . . . 5 x 7 dot matrix . . . . can be set from 80 to 132 columns.
Just think. A Dual Disk Business System for only $3595. Now that's something special.

SNEAK PREVIEW: Hi-Lo-Close CHARTS FOR APPLE II. Moving Averages To Be Announced July 1, 1978

ULTIMATE SPECIAL

Apple II 48K	$1,795.00
Dual Disk II mini-floppy's	990.00
Printer Interface	235.00
Centronics 779 Printer	1,175.00
sugg. mfg. retail price	~~$4,185.00~~
OUR PRICE	$3,585.00
	SAVE $600

ONLY $795

The fully-self-contained Commodore PET Computer System is still ONLY $795.00. Full 8K in RAM, 14K in ROM, full ASCII keyboard, tape drive and 9-inch CRT. Extended Basic is the language, and a machine language monitor is available on tape.

Read from Tape . . . Write to Tape. With PET's 2nd Cassette ($99.95), file maintenance is a breeze.

Now Available! The PET IMPACT PRINTER . . . ONLY $595.00 . . . 60 cps . . . 8½inch paper . . . . YES, it will print the PET's GRAPHICS!

ATTENTION PET OWNERS — New products for your PET MEMORY EXPANSION BOARD to bring your PET up to 32K-$649.95, RS-232 Interface 110-300 Baud (one-way)-$169.95, 110-9600 Baud (two-way)-$295.00.
PET software now available:

MORTGAGE ($15.95)
CHECKBOOK ($15.95)
FINANCE ($12.95)
OPTIONS ($24.95)
STOCK SCREEN ($22.50)
ANNUAL REPORT ANALYZER ($22.95)
STOCK ANALYZER ($34.95)
DATA BASE ($175.00)
BONDS ($9.95)
SUPER STAR TREK ($19.95) SPECIAL $9.95

THE COMPUTER FACTORY

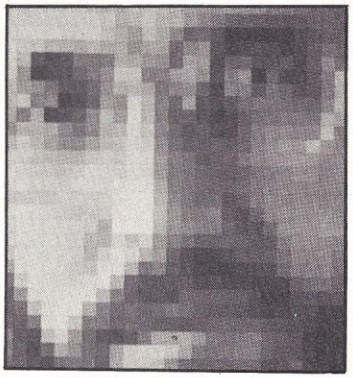
790 MADISON AVENUE, NEW YORK, NY 10021 (212) 249-1666 or (212) PET 2001 T-F 10-6 SAT. 10-4

CIRCLE 5

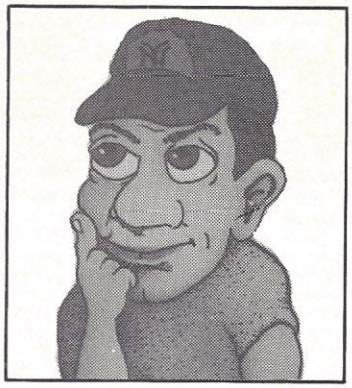
July 1978

VOLUME II, NO. 7

page 25



page 54



page 58

Cover illustration by Dave Gardner

DEPARTMENTS

LAUNCHING PAD

DIGGING IN

IN THE MONEY

ON THE LIGHTER SIDE

FUTURE COMPUTING

ONCE UPON A TIME

INSTANT REPLAY

Personal Computing

JULY 1978 VOL. II, NO. 7

Subscription rates. U.S.: 1 year (12 issues) $14; 2 years (24 issues) $26; 3 years (36 issues) $38. Canada & Mexico: add $4/year for surface mail, $8/year for airmail. In Japan/Asia: Personal Computing, c/o CQ Publishing Co., 14-2 Sugamo 1-chome, Toshima-ku, Tokyo 170, Japan. All other countries: add $8/year for surface mail, $36/year for airmail. Send subscription orders and changes of address to: Circulation, Personal Computing, 1050 Commonwealth Ave., Boston, MA 02215. **Back issues.** U.S.: $3. Canada & Mexico: $4. All other countries: $6.

Publisher. Published monthly by Benwill Publishing Corp.: Harold G. Buchbinder, Chairman of the Board; George Palken, President; Esther Shershow, Treasurer. Executive, Editorial and Subscription Offices: 1050 Commonwealth Ave., Boston, MA 02215. Controlled Circulation postage paid at Long Prairie, MN. Membership in Audit Bureau of Circulation pending.

FEEDBACK

Taxes – Author Reply

Dear Editors:

The following should clarify the three problems mentioned in the Coddington letter, June *PC*:

1. Dividend Exclusion – is not calculated by the program. The user is required to enter the amount of the exclusion as input item No. 2 as shown in Figure 2.
2. The excise tax example was a poor selection. Excise tax is not deductible. As you mentioned, this was a sample input and does not affect the program.
3. Medical Insurance Deduction – again this deduction is not calculated. The user is required to enter the amount up to $150 as item No. 1 and the balance, if any, as item No. 5 as shown in Figure 3. In line 7300 of the code, the program checks to make sure item No. 1 does not exceed $150.

The program is meant to serve as a tool in preparing tax returns and is as accurate as the information supplied by the user. I find it useful as an aid in estimating next year's tax (assuming no rate or tax law changes), since I can use last year's return as the base and modify it as changes become known.

There are income tax software packages on the market selling for $200 that do little more than the package presented in *Personal Computing*. I feel the programs published are a good starting point for a personal tax package and a bargain for only $2 (the price of *Personal Computing*).

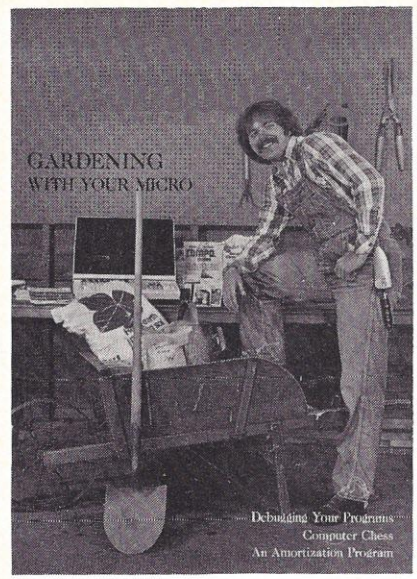
Thanks again for your interest and yes the programs work.

Joseph J. Roehrig
Middle Village, NY

Chess Notes

Editors:

May I compliment your editorial staff? I didn't think it possible to distort my meaning so much with so little effort. I was most impressed with the effect produced by putting Mr. (Bobby) Fischer's name in italics. My letter (April *PC*) now reads as though Mr.



Fischer is somehow associated with the *Chess Challenger*. In the original letter my meaning was unmistakable – the *Chess Challenger* with which I had some experience was the personal property of Mr. Fischer; there was no suggestion that he was in any way associated with the manufacture, production or distribution of that execrable device.

There has also been some misunderstanding regarding the nature of the chess programming I wish to do. I will accept blame – for making assumptions about the reader's knowledge of chess programming, and for my miserable facility with the English tongue.

I do not wish to denigrate the contribution that Mr. Fischer can make to computer chess; rather I wish to emphasize that there are two inter-related tasks in computer chess. One, a programming task, to convert knowledge about chess into programs and subroutines that a computer can process. And two, a chess task, to define the nature of chess ideas in such a way that they may may be formulated into algorithms for a computer.

Considering the current state of computer chess – and I am fully aware of the power of CHESS 4.6 with its huge tree search – the second task is the most important.

Thomas A. Crispin
Goleta, CA

Editor's Note: We are sorry if the italicized name caused any confusion. Mr. Fischer is not related to the manufacturing or distribution of the Chess Challenger.

Market Micro?

Dear Editors:

I recently bought a Radio Shack TRS-80 in hopes of being able to put together a computer display system for personal trading use to receive, store and evaluate real time commodity prices.

Trading information (price, volume and open interest changes) will need to be read off communication lines, then filed on disk storage, run through numerous computer programs (moving averages, point & figure, etc.), results of which are to be filed on disk storage and be available to be visually displayed on a screen in various formats by operator command or automatically by built-in flash alerts.

Because of the recent breakthroughs in computer technology I see no reason why this isn't possible.

Do you know what, if any, additional hardware is needed? Do you know if leasing the real time communication lines will be least expensive direct from the various exchanges (meaning various lines to my system) or would it be better to lease one line from say Bunker Ramo or (?) where desired multiple exchange information would be supplied to my system over one line with minimal time delay?

Being only a part-time investor with limited capital I'm interested in trying to obtain a system at minimal costs, even if it means selling the TRS-80 and purchasing new hardware and software.

Sharon Jackson
Fenton, MO

Editor's Note: Tandy Computer/Radio Shack's TRS-80 *may adequately fill your needs, but you'll need their expansion module, which will allow additional peripherals such as more memory, possibly a printer, and communication interfaces (which we're not certain presently exist in easily interfaced units for the* TRS-80*).*

You may wish to use your micro as both a computer and a terminal. The computer will do all the filing, computing and formatting output; the terminal

will aggregate the data from the communication lines and present it for CRT *display. As a computer, your* TRS-80 *will probably fill your needs if you've provided adequate memory (better to have more than not enough).*

For use as a terminal, we're certain you'll need a modem as the interface between the telephone line(s) and the computer. Also, your computer must switch to a terminal mode or you'll have to use additional equipment (e.g., a TTY *unit which uses punched paper tape and standard teletype supplies). Several models are available from Teletype in Skokie, IL, and Teletypewriter Communication Specialists in Berkeley Heights, NJ; or try a local computer retailer.*

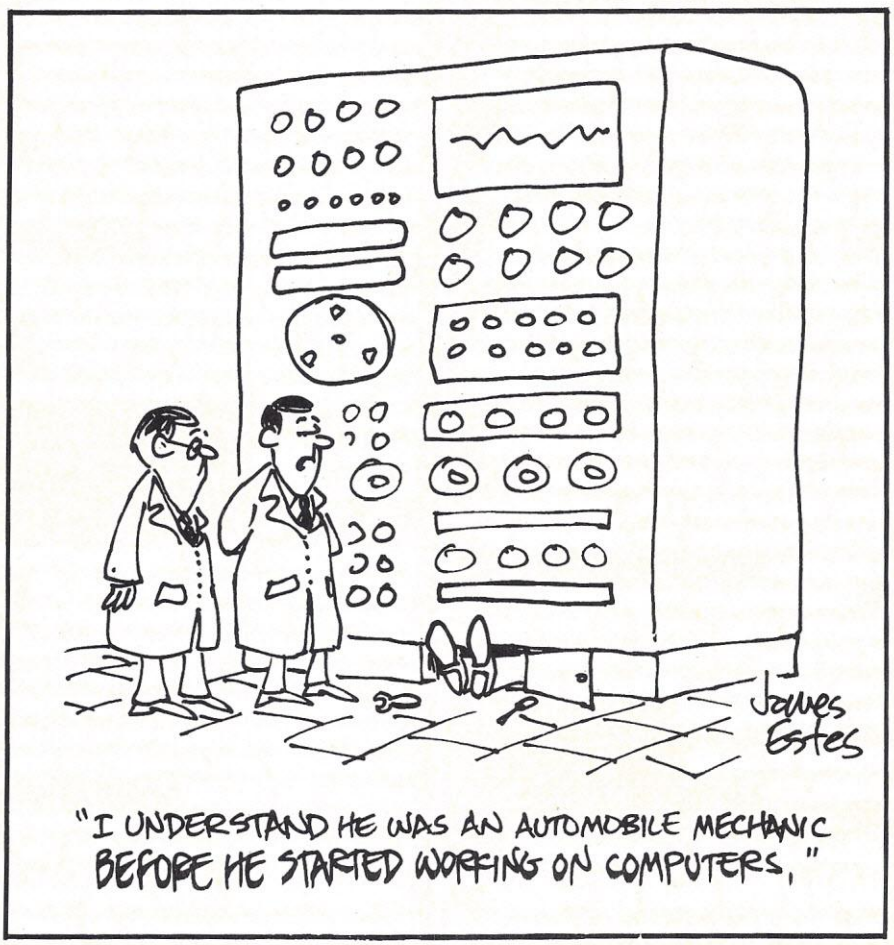
You can buy or lease a new or reconditioned unit. Prices vary, so be sure to review the price structure with the source.

Modems are sometimes included with TTY *units, but may be purchased from Phone 1 of R. kford, IL, or M & R Enterprises, Sunn vale, CA, or your local computer retailer.*

The only other hardware we would recommend is a printer, so you'll have hard copy for reference. If you find the TRS-80 *inadequate for your needs, you might consider Processor Technology's Sol-20/16. It has both terminal and stand-alone computer modes. All you need is a standard TV with an RF modulator (*ATV *research of Oklahoma City offers their Pixieverter,* PXV-2A, *in kit form for $8.50) or a monitor with direct video input.*

Once you have your communications lines linked to a terminal via the interface(s) and you have a floppy disk system in operation with your micro (we recommend a dual disk unit; see March PC *for additional buying hints), your final requirement is software.*

The software could be a problem if you have limited programming experience. It sounds like you'll need extensive software for your application, and we know of none existing that is adequate. Be patient, though; someone may be feverishly creating it now and may contact us or you.



Meanwhile, you might consider Screensplitter video display software which can open your CRT *to displaying both the incoming data and the interactive software on commodities.*

The BASIC *included with floppy disks should be your basis for the programs you (?) will write, modify or commission to write. In this way, your disk file updates and other "housekeeping" chores will be compatible.*

The cost for a complete system as we've described purchased in kit form (although companies are departing from kits) will run about $6000, varying with your final selection. The software could cost an additional $500 to $2000, depending on its complexity, the software writer you use, and whether it is an exclusive package (which you could then turn around and license to others).

The data line (we recommend one incoming line – keeping it simple for software) rental fee is something we can't estimate. Contact the telephone company and your (Bunker Ramo) agent for specifics.

This is the minimum system we recommend. Accomplishing what you want efficiently for less money and equipment (including software) may not allow for expansion as your operations grow. The companies mentioned should be willing to assist and counsel you as part of their customer service.

Ask questions, no matter how inexperienced you think you sound. Take notes and evaluate with a critical eye.

By the time you have all the hardware on line and the software in operation, you will no doubt be a top-notch, electronically aided commodities trader with highly customized programs. Good luck!

Saving a species!

Dear Editors,

I wish to take exception to the comments of James Owen appearing under the heading "Road to Extinction", in the Feedback section of your April issue.

First of all, it has indeed been suggested in the past that program-

mers (but not necessarily programming) might become "extinct" in the then-forseeable future. This particular suggestion first arose in the '50s when the use of compiler languages such as FORTRAN and COBOL began replacing assembly and machine code programming, because the new languages did not require the programmer to be intimately familiar with machine code or assemblers. In this light, the role of the programmer was thought at that time to be diminishing, since the increased ease of programming obviously required less effort – hence less training – on the part of the programmer.

However, with the programmer's task thus made easier, the programmer was then free to become more involved with the aspects of the problem being solved by the computer rather than having to pay such painfully close attention to the coding of machine instructions.

Hence it may be seen today, some 20 years later, that the introduction of compiler languages did not adversely affect the programming profession. Interestingly, the advent of compiler languages for programming purposes actually proved *beneficial* to the position of the programmer. In the (admittedly narrow) example above, the role of the programmer was *changed* somewhat and *greatly expanded* rather than pared into oblivion, as some had expected.

The principal reason for this phenomenon of change and expansion in the field of computer programming was and is that the programmer, particularly the professional, is a not-so-dispensable link in a problem solving chain, or team effort. Problem solving at the professional level involves a great many skills which are not learned overnight, and this is just as true in programming as in, say, engineering.

It is with this idea in mind I point out that the *professional* programmer is not just a mere writer of computer code. To the contrary, the professional programmer is an effective problem solver who must be thoroughly familiar with the fundamental aspects of systems analysis, formal logic and design, not to mention advanced mathematics and communicative skills. These diverse yet interdependent capabilities already required of the programming professional insure that virtually *any* new developments in the field of computing can only strengthen the position of the programmer through change and expansion of the programmer's job function.

Of course, new developments in computing are liable to include programming languages which are easier to learn and use, just like in the previous example. This, however, is hardly a threat to the livelihood of the programmer; rather, it is a boost for the programming business. The reason is quite simple. It is generally true (not strictly speaking, but as a good rule of thumb) that a "simpler" programming language is inherently less versatile and

powerful than its more difficult predecessor. For this reason alone, more difficult (and more powerful) programming languages need to be retained for the successful solution of the difficult problems encountered in professional problem solving.

Even so, it is possible to continue down the path of "degradation of versatility for the sake of simplification" to the point of the software package, which, though so simple-to-use *user* programming is not even involved, is so narrow in purpose and lacking in versatility as compared to a compiler language, assembly language, or machine language, that there must be a special software package for every special application – and the applications are endless!

Such trends in "new" developments are clearly to the advantage of the professional programmer.

In view of the foregoing facts, I found Mr. Owen's analogy between the economics of using microprocessors in control applications (in place of TTL) and the usefulness and adaptability of the programmer to be most amusing. The very example Mr. Owen chose represented one of the hottest new growth opportunities to come along for programmers in years.

As for the reference to "intelligent machines", I seriously doubt that *truly intelligent* machines would be caught dead in the company of humans. After all, as everyone should know, it is *humans* who make mistakes – *not* computers.

Without question, the "Road to Extinction" is riddled with chuckholes!

Mark A. Solis
Baton Rouge, LA

You is correct!

Your (poor) grammar is showing. Your cartoon in the April issue should have been edited so that it was correct.

One *takes* something to someplace that is some distance away, while one *brings* something to the location that is presently occupied.

Best wishes.
Jon Titus
Blacksburg, VA

Oops!

Dear Editors:

Some typos got into the listing of my "Machine Language" article in your March issue. In Program 1 on page 47, LEFTS should read LEFT$ and RIGHTS should read RIGHT$.

Also, HS and XS should have been H$ and X$. In line 320, the BASIC symbol for showing that a quantity is greater than another was omitted.

John Palmer
Tucson, AZ

Catching a Mistake

Dear Editors,

Please note the following corrections – printing errors – in the article "Fishing the MOD Way" (April issue).

On page 53, the last line before the second to last paragraph, the formula should read: 10 J=INT (TR/32) +1 (a parenthesis was missing).

On page 55, Table 1, for Transaction 6, the "Subrecord" should be 3 and "Dummy," 2.

O.E. Dial
Boulder, CO

Write on

Dear PC:

I find your magazine doing an excellent job of tracking this fascinating and expanding field.

D. Bourage
Vienna, Austria

We Goofed!

Much to our dismay, we discovered that our May cover story, "If It's Tuesday, This Must Be Seattle", was printed without its by-line. The story was written by Sam Newhouse. We truly regret the omission. – The Editors.

RANDOM ACCESS

Electronic "toys" - a child's game?

Electronic and video games and "computer toys" are selling like hot cakes — just ask Milton Bradley, Parker Brothers, Mattel or any one of the toy manufacturers who have stepped into the growing electronic game and toy field.

"Kids today are more sophisticated than those of yesteryear," commented one toy store manager, "and so are the toys they choose to play with."

This year alone, one manufacturer, Coleco, introduced five new low-cost microprocessor-based games. The games are simple to learn and to play and they appear (to us) to be suited for "kids" of all ages.

Coleco's new Electronic Quarterback (no it's not a Joe Namath doll you plug into an outlet) offers two skill levels, Semi-pro and Pro. The player directs the offense and a microprocessor takes care of the defense. Players can run, kick, pass and block, all to built-in sound effects. LEDs display time, score, downs, yards-to-go and field position.

Other Coleco games include Quiz Wiz, Zap (an LED missle-race game), Amaze-a-tron, and Digits (their version of Mastermind).

Promoted at the 1978 New York Toy Fair as "a game so sophisticated it has a voice, a language and an intelligence all its own" was Parker Brother's MERLIN. Suited for two to four players 7-years-old and up. MERLIN offers a library of six computer games — games of chance, strategy, memory, logic and skill from basic tic-tac-toe to "Mindbender".

MERLIN counters (with a 2K-byte memory) the players' moves with its own strategic maneuvers and communicates with a vocabulary of electronically synthesized sounds.

Milton Bradley's latest electronic "brain teaser", Simon, generates random sequences of colored lights and sounds the players must memorize and repeat.

With so many companies entering the electronic toy market, it's beginning to resemble the microcomputer industry. Companies are fighting to introduce a new product first for fear that someone will undercut their price or produce a better product.

For example, *Electronic Engineering Times* reported that just days after Coleco showed its low-cost programmable video game at the 1978 Toy Fair, Atari, the only other manufacturer to come up with "less sophisticated and less costly versions of their own ROM programmable video games," confirmed that its low-priced programmable unit would not be marketed.

According to *EET*, Atari found its low-cost game did not do well in relation to the company's highly successful microprocessor-based Video Computer System. Michael Shea, director of marketing at Atari, was quoted as saying, "Price elasticity doesn't seem to be as major a factor as play value and game sophistication. The consumer keeps voting for the microprocessor-based product."

Sound familiar? Maybe it's not just child's play. . .

— *Ellen Stein*

Illustration by Nancy Lawton

Game of Kings

Another version of "the King of games and the game of Kings" — Gammonmaster II — has hit the shelves. Marketed by Trycom, Inc., the new model of the 200 year old game utilizes microprocessor technology to produce a self-contained, portable backgammon set.

Programmed to recognize and defend itself against all strategies of the game — running, blocking, blot hitting contest, back and semi-back games, bearing off strategies and combinations of these basic techniques — the set adheres to the rules of classical backgammon.

An electronic rolling of the die ensures randomness of play and uniqueness of games.

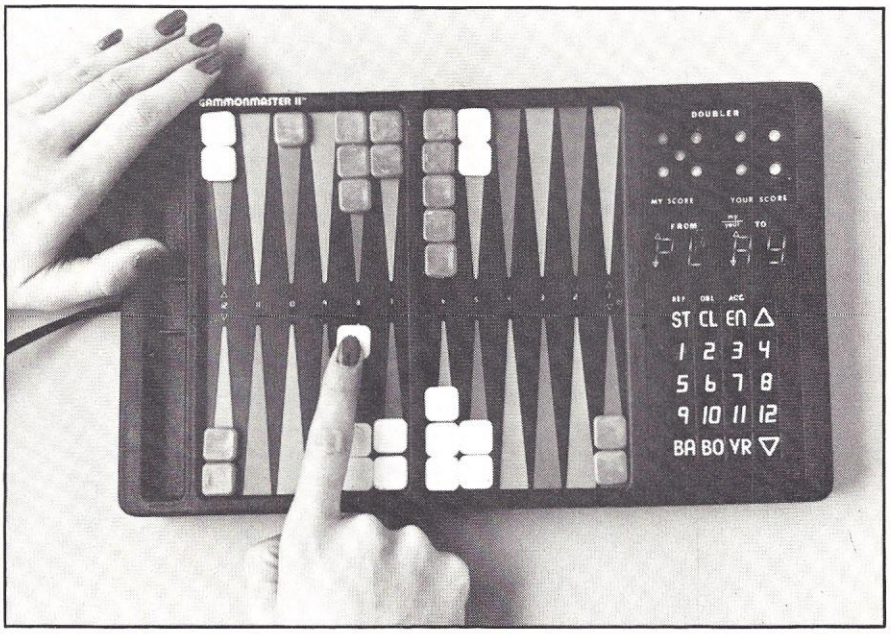
The electronic doubling cube, available on the higher priced "G-DOUBLER" or as an upgrade to the original model, adds to the game through the use of betting, controlled by tournament play regulations.

Backgammon is a simple game to learn, yet difficult to master. Games are brief, fast-paced, and call for a degree of skill, knowledge of the basic strategems and the luck of the dice.

The electronic version competes evenly with advanced players, improves the intermediate player's game and teaches the novice the techniques of backgammon.

The company will also market a coin-operated cocktail lounge verson of the game as well as CARDMASTER, an electronic card partner that plays gin, poker and bridge (scheduled for summer production).

For more information, contact Tryom, Inc., 23945 Mercantile Road, Cleveland, OH 44122.



Hands on computer courses

A series of hands-on microcomputer courses offered by Tycon, Inc., Blacksburg, VA, starts with a three-day Microcomputer Interfacing Course (No.628) at the Personal Computing '78 show, August 22-24 at the Philadelphia Hilton hotel. This course emphasizes the use of 8080-based computers, with over 15 development systems available for the students' use. Almost one half of the seminar time will be devoted to lab experiments covering software and hardware interfacing.

Three microcomputer courses will also be offered by Tychon at their Blacksburg, VA, Learning Center, starting with Designing with Microcomputers (No. 626), September 11-13. In this course, participants will design and build their own microcomputer, based on the 8085 processor. Each student's computer will contain a serial I/O port, parallel I/O ports, a timer, and PROM and R/W memory, containing DBUG, a monitor program.

The Microcomputer Interfacing Course presented in Philly (No.628) will be repeated Sept. 14-16, covering the basics of I/O hardware and programming.

A new, five-day Software Design course (No.690) will be offered Sept. 18-22. The course teaches 8080/8085 assembly language programming, interrupt programming, data manipulation, code conversions, editors and assemblers.

Students will edit, assemble, debug and test programs that they develop as a part of the laboratory experiments.

All courses taught at the Learning Center incorporate special experiments which include a traffic light controller, electrostatic printer, logic analyzer, IC tester, phone dialer and ASCII keyboard.

Course fees range from $295 to $495 and include books, manuals notes, lunches and coffee breaks.

For more information, contact Dr. C. Titus, Course Director, Tychon, Inc., P.O. Box 242, Blacksburg, VA 24060.

For the do-it-yourselfer

What's available in electronic kits for the do-it-yourselfer can be found in the spring Heath Company 104-page catalog. Kits for amateur radio, high-fidelity components, test instruments, digital clocks and personal computer systems are listed.

New products in the catalog include games software for the H8 computer system.

For more information write to: Heath Co., Dept. 350-590, Ref PC, Benton Harbor, MI 49022.

Blackjack, the easy way

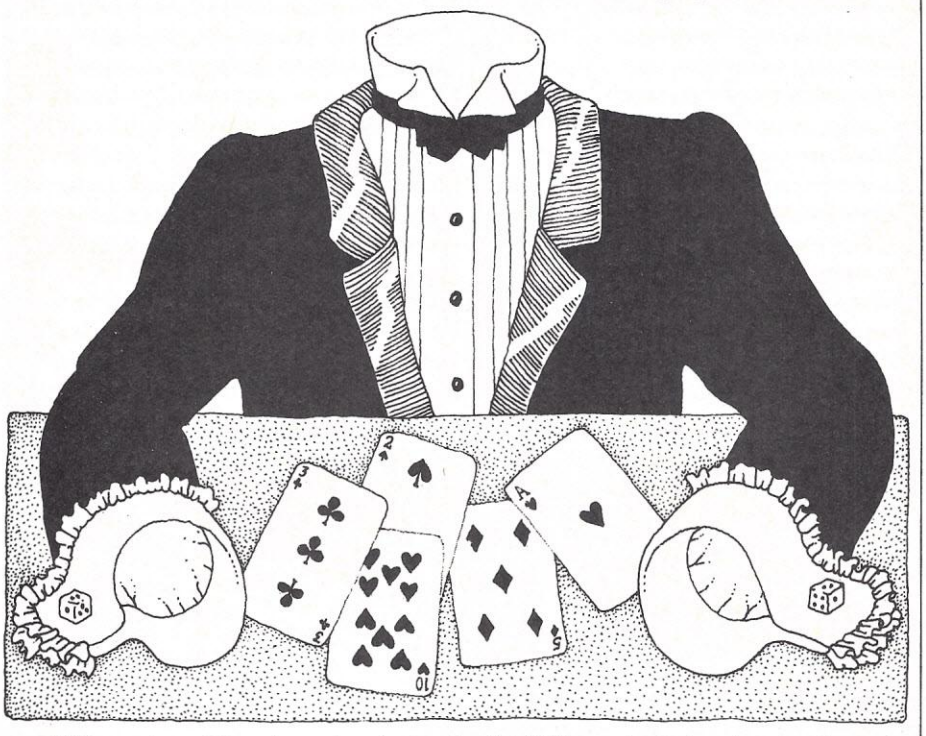
What? A Las Vegas blackjack dealer who never cheats, doesn't wear a tuxedo and won't take one penny of your money? Impossible. Or is it?

Welcome *PLAYBJ*, a *BASIC* computer program developed by *Gambling Times* Magazine for blackjack enthusiasts and computer game players.

Before, a serious black-jack player had to test his skills in the casino and face the risk of losing his entire bankroll not to mention his shirt. Now, he need only face his *CRT*, which reveals his hand and his bankroll in a fixed-screen display. The player specifies his wager and as he wins or loses, the total fluctuates accordingly.

Developed as a teaching device under the direction of blackjack expert Stanley Roberts, *PLAYBJ* stimulates the precise game played in the Las Vegas casinos.

While many blackjack programs available do not offer all the Las Vegas options and so limit a player's ability to use the program for sharpening his casino skills, *PLAYBJ* allows: splitting up to four hands during the deal, doubling down after splitting and insuring the player's hand when the dealer shows an ace. It also lets the player stand, hit till he busts or surrender. In addition, a choice of from one to four decks may be used in the shuffle so the player can practice the multiple-deck game used in most casinos.



Although written in extended *BASIC* for a Jacquard model J-100 minicomputer, the program can be modified to run on any computer (micro or larger) which uses *BASIC* and has a *CRT* capable of printing characters at given coordinates. The program requires 16K to 20K *RAM*.

Gambling Times (839 North Highland Ave., Hollywood, CA 90038) offers the program which includes listing, documentation (explaining each 5 to 10 lines of code) and modification instructions for changing from Jacquard *BASIC* symbols to standard *BASIC* for $10. For those with the Jacquard system, they'll sell you the floppy disk for $50.

So if it's a bad day and you're losing money, you needn't worry. Just turn off the display terminal and try your luck tomorrow — no poorer than when you started.

What's Whatsit? And who's on first?

WHATSIT — or, "Wow! How'd All That Stuff get In There!" — lets your home or business computer manage a "data base" of stored information and respond to your queries in pidgin English.

First demonstrated at the West Coast Computer Faire, the compact system runs in BASIC yet it brings the power of a data base manager to the business or professional user.

You can index your music collection, computerize your customer list or keep tabs on your investments. With WHATSIT you can do these things as well as many other chores that demand quick access to random disk files.

You retrieve or store data by typing pidgin English requests, such as: "WHEN'S DR. JEKYLL'S APPOINTMENT?" Indexing and disk space allocation are handled automatically. Used as a business data base manager, WHATSIT can index data.

File structure is never frozen, but develops automatically through normal use to adapt to user requirements. Stored information is automatically cross-indexed under desired headings. Headings may be added or changed at any time by pidgin English requests; no reprogramming needed.

Available in North Star BASIC, the system runs in 24K of memory. WHATSIT is offered complete ($75.00) with three ready-to-run programs on a minidisk plus an extensive manual written in non-technical language.

Also offered separately ($25.00), the manual includes program listings, user instructions and a beginner's shopping guide to inexpensive computers.

Check your local computer store or write: Information Unlimited, 698 W. 70 So. Private Rd., Hebron, IN 46341.

Illustration by Nancy Lawton

In review

How Small Businesses Use Computers, by the editors at Management Information Corp.; MIC. Cherry Hill, NJ. $15 paperback.

In *How Small Businesses Use Computers,* Management Information Corp.'s editorial staff put together 15 case histories that originally appeared in their newsletter, *Small Business Computer News.*

Edited and written in a conversational and matter-of-fact style, the selections give some basic facts about the particular company and then trace its path toward installing a computer.

Histories were drawn from businesses including: hospitals, manufacturers, nursing homes, wholesale/distributors, local government organizations and bureaus.

Each entry follows the same basic format. MIC's editors provide the company name and location, the type of business and where appropriate: employee number, sales volume, DP personnel (how many and what types of employees), DP budget, hardware configuration and software. Not all headings appear with each entry.

After this brief introduction, the editors present an informal case history during which they occasionally digress to point out situations from which readers can learn. At these places, the editors sometimes can't resist giving advice.

For example, when discussing a service bureau's minicomputer installation, the type of services it provides, its customer types and how much they pay, MIC's editors suggest service bureau customers begin looking at an in-house computer when their monthly bill exceeds $1000.

Although each case history does not carry identical information, basically they all describe how each business's computer system evolved — what worked, what didn't, what plans are for the future (in both hardware and software terms), what type(s) personnel operates the system and any other changes that might have taken or might still take place.

One history might explain who selected the system and how, while another might just mention that the need arose for a computer and so company X installed one — but taken together, the sections present a wide variety of experiences to describe problems potential computer users might run into.

Since "typical" cases or "average" cases usually miss something — because no single installation will run into every possible, positive or negative, experience — presentations of multiple individual cases can hit almost every calamity or success an installation might see.

All in all, it is an informative, effective and easy-to-read book.

— H. Paris Burstyn

"SOME SAVER; IT ELIMINATED 50 EMPLOYEES AT 3 DOLLARS AN HOUR AND REPLACED THEM WITH 15 REPAIRMEN AT 10 DOLLARS AN HOUR."

Unlocking communications

An electronic communicator for cerebral palsy victims and others whose loss of speech has trapped them within their handicapped bodies has been developed at MIT's Sensory Aids Evaluation and Development Center.

The device uses a normal television set and a microprocessor that can be tailored to the widely varying needs of the

handicapped by selecting one of many stored computer programs.

The MIT Universal Communicator (or Unicom) can be operated through many input devices, ranging from a single switch which requires only a slight movement by the handicapped person to a full typewriter keyboard. Closing the switch transfers selected letters, numbers or words from the selection matrix shown on the television to the message field of the screen where messages may be edited before being printed.

The ability of a user to edit the text being prepared is an important feature overlooked in many other communicators according to Dr. Derek Rowell, director of the center. The facility to delete, change or insert characters in the previously formed portion of the message is important.

The Unicom also can give the handicapped a way to control their environment. For example, turning on lights, radios, televisions or other electrical devices, calling a nurse or controlling heat or air conditioning equipment are possible by switching to an appropriate program.

Another application of the system is as a self-paced teaching aid for handicapped children.

According to Dr. Rowell, the device is easily adaptable to the needs of each individual.

If all the handicapped person can do is nod his or her head a switch, activated by the head, can be rigged up.

Unicom can be used in several modes. Individuals with the greatest degree of motor control can use the communicator in the "direct selection" where the user directly identifies the portion of the display to be sent to the message field of the television screen.

The "encoded mode" is used when the handicapped person has less motor control. The user must follow a coded sequence of actions to move letters to the message area.

In the "scanning" mode the communicator is suitable for those who are capable only of activating a single switch. The words and characters are displayed in eight lines of 16 characters each. A cursor moves along the lines. The first motion of the switch selects the wanted line, the second motion the character on that line which is to be sent to the message field.

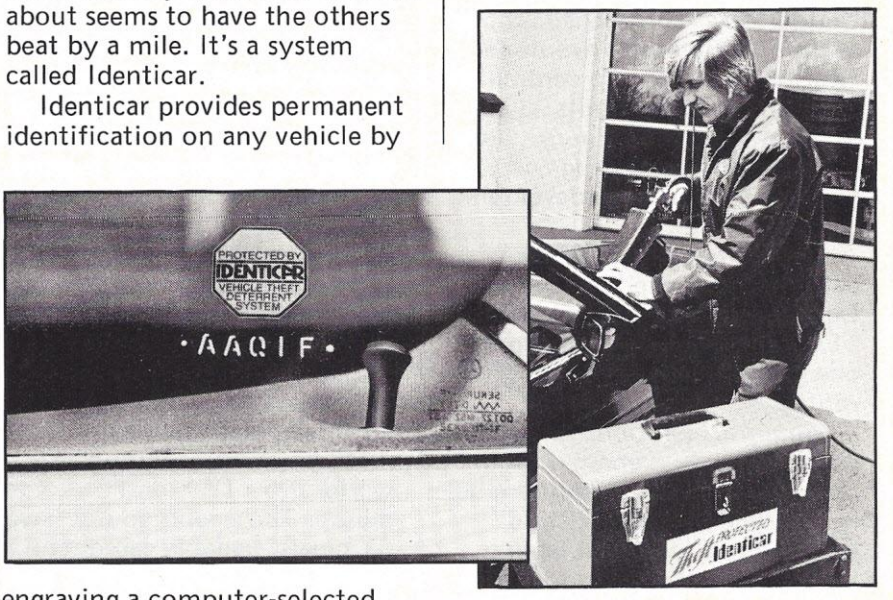
Before settling on a design philosophy the MIT engineers surveyed existing systems and discovered a tendency toward simple devices that are inconvenient to use.

The survey also found little evidence that such devices were designed with flexibility in mind, one reason why earlier devices have not come into wide use in hospitals or rehabilitation centers.

Computer car codes fight crime

Theft-proof locks, burglar alarms, kill switches, "hot car" stickers — all kinds of gadgets exist to protect your car from theft. But the latest security device we've heard about seems to have the others beat by a mile. It's a system called Identicar.

Identicar provides permanent identification on any vehicle by engraving a computer-selected code into every piece of glass.



Because of the physical properties of automotive glass, the code cannot be altered or removed without total glass replacement at an average cost of $700 to $1000. So, since the thief knows he cannot resell the vehicle, he'll look for another car to steal.

The computer code consists of a series of small (3/8-inch-high) letters, and each vehicle is assigned its own unique marking. All codes, together with other pertinent information such as vehicle description, owner's name and vehicle identification number, are maintained in a national registry. The information is available to law enforcement agencies via a nationwide toll-free telephone line on a 24-hour, 7-day-per-week basis.

Thus, in addition to deterring theft, the code facilitates recovery and identification of stolen vehicles. Any Identicar-protected vehicle can be traced within minutes from anywhere in the United States.

Unlike other anti-theft devices, Identicar cannot be bypassed — you don't have to remember to activate an alarm, engage a lock or throw a switch.

Though the system was developed to deter automobile theft, it works for any vehicle with glass, including mobile homes, boats, buses or trucks. The code can be engraved while you wait.

What have you got to lose?

Patent Service for Inventors

A worldwide computer service for individual inventors in the United States and abroad, known as the International Inventor's Registry, consists of a centralized data base, listing patents issued in the areas of solar energy, agriculture, food processing and technologies of benefit to developing countries.

The listing service ($100 per patent), announced by Control Data Corporation as part of their TECHNOTEC service, will offer one year listings of patents to inventors.

Presently, solutions to global problems such as energy, food, water and unemployment are being delayed because existing knowledge is locked in laboratories and workshops of individual inventors. Developers hope the patent service will facilitate technology transfer to help solve these problems.

To subscribe to the International Inventor's Registry, inventors must have a patent issued in the solar, agriculture, food processing or other appropriate technology areas. They may then contact Control Data in Minneapolis, or any Control Data representative throughout the world, and arrange to record their invention on the computer-based registry.

Inventors interested in the service must hold a current patent from the U.S. government or a foreign country. If granted abroad, their patent must be accompanied by an English translation.

Users can research the data base via computer terminal, using local telephone connections in most major cities throughout the world or by TWX or TELEX terminals. Those who do not have terminals available can obtain a search from TECHNOTEC and receive results by mail or telephone, the company said.

Don't let your eyes fool you. You're not looking at Zenith's, Sony's or Motorola's manufacturing plant. The picture above shows the final production stage of the TRS-80 at its Fort Worth, Texas, plant. Here, the micros receive their final burn-in and quality assurance check before being shipped out.

Computer portraits

A Chicago group of home computer hobbyists have developed a new computer portrait system. Built from the ground up, the Chicago system incorporates the popular Z-80 processor chip, specially designed analog/digital converter and other special purpose circuit boards, all of which plug into the S-100 bus.

While designed primarily to make computer portraits, the computer has sufficient memory and capability for performing general computer applications with high speed hard copy on its Centronics 102AL printer, the group says.

A new portable version will soon be marketed, and special purpose boards and kits will be marketed separately so hobbyists can build a similar system on any S-100 bus computer.

In addition to outright sales with an optional service contract ranging from $7000 to $10000 for the basic system, the company will become a partner in the system if you have a good, high traffic location.

For further information contact: Digital Enterprises, Inc., 6033 N. Sheridan Rd., Suite 36K, Chicago, IL 60660.

Service Directory

An updated version of the *Directory of Computer Based Services,* published by Telenet Communications, offers a complete listing of interactive computing and information retrieval services. More than 80 data banks, commercial service bureaus, education institutions and other companies are categorized by application specialty, programming language and data base offerings.

Besides a variety of bibliographic data bases, the directory contains listings of data bases in other areas, including financial and economic data bases, FCC tariffs for communication common carriers, school guidance information, advertising media and market research, energy and pollution,

and engineering.

Specialized computing services available to the public through these subscribers are listed in five categories: general business, engineering, special industry, scientific and education.

Copies of the 1978 directory ($2 each prepaid) are available from: Publications Department, Telenet Communications Corporation, 1050 17th Street, NW, Washington, DC 20036.

6800 users gather together

Not all new hobbyists groups are for PET or TRS-80 owners. For example, a 6800 Users Group has just been formed in the Dallas/Fort Worth, TX, area.

Currently, the group discusses varied topics of interest to users of 6800 systems, offers an assortment of presentations and works on tutorials in assembly language programming.

Those not in the Dallas/Fort Worth area, still can take part in the group's activities through the "Ask the Chips" portion of their meeting. Here, any questions, problems or comments you might have concerning 6800 systems can be presented to the club and discussed. Any user worldwide may correspond. A club spokesman said they make every attempt to respond with any solution(s) they might have discovered.

If you're into the marketing end of computers, as a manufacturer, software supplier or software writer, or if you have formed a club of your own, you can get in your two cents worth through their "Tell the DIPS" part of the meeting. The group will distribute your brochures among the group or you can address the club directly.

The group meets on the third Thursday of each month, 7 p.m., 1220 Majesty, Dallas, TX. For more information write Charles Matz, 4114 Avondale, Suite #2, Dallas, TX 75219, or phone evenings (214) 522-7130.

"Debugging" by computer

Budworms, bollworms and weevils better run and take cover, 'cuz the computer is gonna get them!

Scientists and agricultural specialists at Texas A&M University are developing a new weapon for the farmers' pest-fighting arsenal — "Bugnet," a computer-based pest management program.

By adding field data to the IBM computer-based insect, plant and economic models developed by the Bugnet team, Texas agricultural extension service specialists will be able to help farmers plan strategies for more prudent uses of pesticides, thus eliminating costly guesswork, according to Dr. Don W. DeMichele of the university's biosystems research division.

One other advantage of Bugnet is faster information availability, explained DeMichele. "Using conventional means, such as publishing our findings, it could take 10 years before the information is of any value to the farmer."

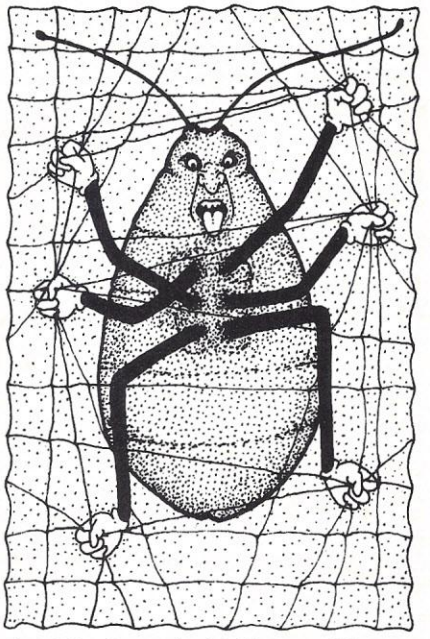
Many problems that plague farmers show the need for a program like Bugnet. For example, a certain degree of, say, fruit shedding from a cotton plant is natural; but a farmer needs to know how much of what he finds *is* natural and how much is caused by pests.

Another problem is the time element. Boll weevils, for example, are a common cotton pest and can increase their number by a factor of 100 in one generation. Thus, farmers must apply pesticide within the span of a few days to be effective.

"It's a fragile system," DeMichele explained, "You've got to get the adult weevil after it leaves the protection of the cotton bud and before it begins to lay eggs. But when you spray the pesticide you also destroy the natural enemies of the bollworm, and alternative measures must be taken for this pest." That's where Bugnet comes in.

To develop the project, the Bugnet team uses eight IBM 5100 portable computers. DeMichele estimates that in three years this type of small computer will be in offices throughout the Texas Agricultural Extension Service and "user-oriented" computers for extension specialists requiring only a half day's training will also be available.

Bugnet will be tested this spring on cotton crops in the Blacklands, north of Dallas, in



the Rio Grande Valley and in the High Plains of West Texas.

The pilot effort will include weather forecasts; insect alerts; crop budget analysis; market outlooks; bollworm, budworm and boll weevil population predictions; cotton plant models and sophisticated insect control decision aids.

If successful, DeMichele said, extension specialists will spearhead Bugnet's information delivery system to benefit growers of all major crops across the state.

Producers are caught in the dilemma of rising costs and dwindling resources such as energy, fertilizer and water he explained. To cope with this, they must become more efficient, and the information provided by Bugnet's computer simula-

Illustration by Nancy Lawton

tions of agricultural production will help.

Because Texas is such a large state, farmers in many localities don't have access to a computer. With Bugnet, they should have access to a resource they've never had before.

So all you weevils out there, LOOK OUT!

Pet Papers

Whether you're a beginner just training your PET, a knowledgeable PET handler or a PET "vet", you'll find something of interest in a new publication entitled the "PET PAPER".

Published by ex-Commodore software coordinator Terry Laudereau and KIM product manager Rick Simpson, the coast-to-coast newsletter will offer something for everyone.

Editorial content will include lists of known user groups (so you can exchange information locally), methods of starting your own user group, articles on PET BASIC, hardware and software notes, reviews of all known and available cassette programs, sources and costs of PET peripherals and accessories as they become available and comments and condensations of nationally published articles of interest to PET owners.

At present, the PET PAPER is also offering a flea market software exchange library. The list includes such games as Othello for one or two, road race, slot machine and an address file program. You just send them a tape of a program you've written and if they accept your program you'll receive a coupon for one of the programs in their library (you do pay $2 for duplication costs and postage).

Or, if you don't have a program of your own to exchange, they'll send you a program for $5.

For subscription or cassette exchange information, contact the PET PAPER, P.O. Box 43, Audubon, PA 19407.

Buying guide

The Spring 1978 Byte Shopper provides a key to computer system selection and a guide to disk drive selection. New sections focus on systems designed specifically for business applications.

The Shopper also includes an introduction to personal computing with a glossary of computer buzzwords, and graphics of how microcomputers work and where they can be useful. Over 125 photographs show equipment from more than 120 manufacturers.

Manufacturers' specs are combined with a discussion of how to use each product and how it relates to an overall computing system. Several "typical" systems from simple home systems to large timesharing multi-user systems for business are pictured and discussed in detail.

An 8-page price list has been included for prospective buyers. For more information, write to: MicroAge, 1425 W 12th Place #101, Tempe, AZ 85281.

Gulf Coast group

For those of you in the Gulf Coast Area who have been waiting to join a computer hobbyist group, your wait has ended.

G^2C^3 has just been formed in Mobile, AL. Their present membership ranges from novices to established professionals, ensuring a wide variety of subjects for discussion.

Anyone in the Gulf Coast area is invited to join them. Meetings are held the first Wednesday of every other month.

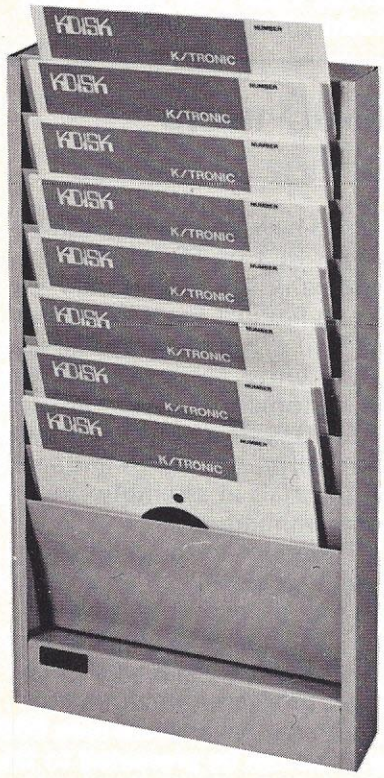
Members often bring their computers to the meetings for demonstrations and presentations on building techniques.

For the location of G^2C^3's next meeting and any additional information, call (205) 478-1777.

Filing floppies

If you improperly handle or store your diskettes, they won't last long.

File boxes, storage racks, folders, record album holders — whatever means of storage you choose, you're better off with some form of protection than with none.



A relatively new item on the market is the Floppy Disk Hanging Storage Rack by Robins Industries ($27.80). Designed for wall or desktop mounting, the rack stores up to 16 floppies, 2 in each pocket. The front plate has room for holding pencils, clips and other items.

The rack measures 18" x 10" x 2" and is finished in white enamel.

For more information contact Sam Jones, Sales Mngr., Robins Data Products Div., PC#78-053, 75 Austin Blvd., Commack, NY 11725.

If you have a photo of your system you'd like to share with our readers, send it to Random Access, Personal Computing.

To buy or not to buy: is that the question?

BY DAVID PRICE

You've heard computer horror stories before. The myriad computer fraud cases, for instance. Or the programming error that caused the destruction of a multi-million dollar satellite. They're not too hard to find.

If you own a small business computer (or are contemplating the acquisition of one), you'll want to protect your business from the devastation of such an incident. In fact, you'll probably need to be even more circumspect than your counterparts in the big-computer world.

Why? Because the microcomputer market has historically emphasized price competition rather than customer support. Which is all rather ironic, since the small business computer owner needs this support even more than his corporate cousins. Generally, small businesses cannot afford to hire expensive specialists, do not have the expertise in-house, and are less capable of absorbing the concommitant financial loss. In other words, it pays to be prepared.

The purpose of this article is to better equip you for this preparation process. Moreover, it will provide a jumping board from which you can develop your own evaluation criteria.

This article contains many questions, but very few answers, because *you* must consider and answer the questions as they apply to your situation. And you shouldn't be afraid of "undesirable" answers. Remember: you aren't doing yourself any favors by pretending a pitfall doesn't exist or by counting on it going away. Only careful, logical planning will eliminate potential problems.

The Suppliers

If the suppliers are not reliable, a multiple of problems can result. "Suppliers" refers to both the manufacturer and the retailer. They are almost always independent organizations, and both of them should be ready to back up what they sell.

First, consider the manufacturer: does he take pride in his products, or does he seem to be after a quick buck? How long has he been in business? Is his a national organization, or does he operate out of his basement? In short, can you trust him?

Fortunately, incidents of outright fraud are few and far between. What does occur with disheartening frequency are situations where the manufacturer starts with the best of intentions only to discover he simply cannot, for any number of reasons, deliver the product. (He might, for example, find himself unable to secure a certain scarce component.)

Another (infinitely more insidious and not uncommon) setup occurs when a company solicits orders for a product which, unknown to the buyer, doesn't even exist. Why? One explanation was printed in the November 1975 issue of *People's Computer Company*:

"Insiders in the computer world learned a long time ago that things aren't always what they seem. And it isn't just that products aren't always what the advertising says they are: sometimes there isn't any product at all. This can be deliberate. It has been known for a company to announce a new machine in glowing terms, although it did not exist, solely as a maneuver intended to inhibit potential purchasers from buying from a rival. Where this was a straight bait and switch ploy, deliberate intent is obvious. Sometimes the intent may be more honest although the effect is the same. Markets, by definition, are competitive and the timing of the announcement of a new product may be critical to its success. The electronics industry is particularly sensitive to this . . . "

How, then do you avoid getting burned? Here are a few common sense things to remember.

Read up on the product. Find out if it has been reviewed by a magazine, trade journal or club newsletter. Is the article an objective one? If it seems to consist mainly of the manufacturer's puffery, take its claims with a bit more than the proverbial grain of salt.

Ask other hobbyists about their experiences with the product – and the company. (Computer club meetings are excellent for this.)

Study the product manuals. If you can't borrow them from somebody, it might pay to buy copy(s). Most manufacturers will sell you their manuals separately. Find out how well written they are. Note whether they're professionally laid out and typeset, or if they were run off on the office copier. Product manual documentation often offers a clue concerning how conscientious a company is in other areas, as well.

Always remember, you should not be basing your deci-

Illustrations by Nancy Lawrence

sion on one source alone; gather information from a broad range of inputs.

The Retailer

Even after you're confident you're dealing with a dependable manufacturer, ask the same set of questions about the retailer. After all, quality equipment won't do you much good if your dealer isn't helping you use it properly.

Some people need more hand-holding than others – how much will you need? And how much can you get?

How many of your personnel will need training? Will the retailer you buy your system from train them? How will they be trained? Will they have to attend classes at the retailer's outlet? Will they have to study at home?

Consider durability. What kind of warranty backs up the various hardware components? The software? When the system needs maintenance, how will it be performed? Will you have to bring it in to the outlet, or can you get on-site service during office hours?

And what about preventive maintenance? How long will you have to wait for repair service? How long can you afford to have your operations disrupted while waiting for it to arrive? What will the situation be after the warranty expires? Is a service contract available?

Make sure you get a firm commitment from your dealer in terms of post-sale support. And even though the prices

may be lower at the supermarket-type dealer, consider whether you might be better off in the long run by paying a little more for an assured higher level of support.

Evaluating Your Needs

Before you lay down your money for a system, make sure the one you've chosen fits your needs. If the system can't be integrated into your operations and do its job efficiently, it may become an underutilized, overrated, expensive nuisance.

Consider how much staff time will be spent collecting data and preparing it for input. How will this preparation and input be accomplished? Are the procedures workable or will you need to hire additional personnel? Can you see any bottlenecks forming as a result of time-consuming procedures or processes?

When justifying the existence (expense) of your proposed system, consider the output – just how useful will the information generated by the computer be? Will its frequency suit your needs? What will you be able to determine from this information? To what extent will your short- and long-range strategies be affected by your analysis of it? And taking all this into account, do the benefits of the system justify its cost?

Also keep the future in mind – you wouldn't buy last year's car if this year's model got better mileage for less money. So consider the possibility of system expansion. Does your system allow it, or will you be frozen into one configuration? How easy is it to add to main memory, mass storage and other peripherals, such as input and output devices?

And your software is just as important as your hardware. How expandable is your software? Does the documentation give you the information necessary for later modifications? Can the manufacturer or dealer make these changes for you? At what cost?

The final question concerns the danger of obsolescence. That is, will peripherals, software and service continue to be available for your system? Is it extensively second-sourced? Does it use a standard architecture – such as the S-100, or Altair, bus design – which is supported by many manufacturers? If so, you're on much safer ground than if you commit yourself to an unpopular processor or a nonstandard bus structure. This should be pretty obvious: if your system uses a common architecture, and something happens to your original supplier (heaven forbid), you save yourself from the disaster of being stuck with a machine that *nobody* is supporting. Also, by using a widespread architecture, you wind up with a much better selection when it comes time to buy new equipment.

Don't forget your customers. How will they react to computerized material? Will they benefit from the system, will they be alienated by it, or will they even care? Do the benefits of the system outweigh the risk of depersonalization?

The Options

You don't have to buy a system even if you decide your company is ready. Consider the options available. Many dealers offer lease plans, for example. Timesharing, another alternative, lets you access a remote computer via phone lines, paying only for the terminal time, processor activity and memory space that you actually use. As the name implies, a timesharing system handles many users simultaneously; it responds so quickly, however, that for all practical purposes each user has the functional equivalent of an entire computer at his disposal.

Don't forget the possibility of an outside service bureau either.

All three of these approaches have two rather obvious advantages: first, your money is not tied up by a high initial outlay, and second, because you don't actually own the equipment you're using, you remain free to terminate your relationship with a particular vendor with relative ease.

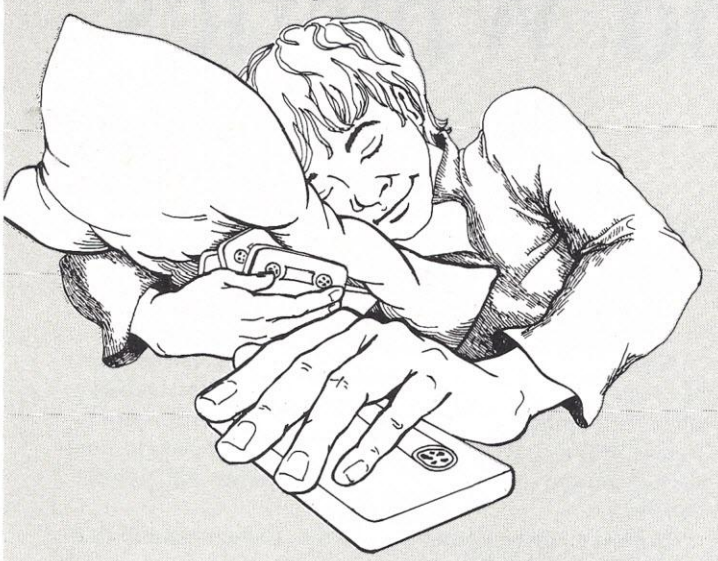
To summarize: you must decide whether the system is well suited to your application, whether its resources are sufficient to cope with the demands of your application and

whether it will disrupt or enhance your operations.

Privacy

Most business applications of computers involve record keeping. In fact, record keeping is usually one of the computer's primary functions. In this area you should think about a number of potential problems.

First, and most obvious, how reliable is your equipment? A system is only as strong as its weakest link, after all. Does your main memory use some sort of error-checking scheme? Ditto for your mass storage devices: how reliable are they?



Remember that no such device is 100 percent error-free; the best you can do is minimize the number of errors.

Current equipment designs have remarkably low error rates, though. Your primary concern should probably be the storage medium itself – the magnetic disks and tapes. Don't skimp on them.

Unless you're prepared for it, a power failure – even a quick brownout – can cause a great deal of harm. If they're common in your area (particularly during the storm season), make sure you're protected. Commercial systems frequently include special power supplies to protect the system from power fluctuations. And, in case of a total blackout, some go one step further and automatically shut themselves down, thus preventing loss of the data in memory (as well as damage to sensitive hardware).

But, in spite of all of your precautions, your records may wind up damaged anyway. The magnetic radiation from your janitor's vacuum cleaner could erase everything on your main disk. What then?

Once again, an ounce of prevention is worth innumerable pounds of cure. One useful practice is to maintain a "backup disk" (or tape). This simply involves creating duplicates of your disks, and storing the copies in a safe place. You might do this at the end of every day, for example. Then, if catastrophe occurs, it won't really be that much of a catastrophe at all, because the only information you'll lose will be that which was placed in the disks since they were last duplicated.

People Problems

Now that you're better protected against hardware and software failure, consider another possible source of error: people. How do you eliminate human error? The answer is, of course, you can't. But you can build safeguards into the system to minimize it.

The very best safeguard is a competent user. Who will be operating the system, or feeding it data? If they're familiar with the usage procedures, you're that much better off. If the general public is to use the system (via public terminals of some sort, or what not), make sure the usage instructions are easy to understand.

Consider the inputs that will be made: do the input-handling sections of the software have safeguards against invalid inputs? Here are a few typical examples:

1) An order number or transaction code. These generally have a fixed number of characters. One useful error-checking scheme would be to make sure that all such inputs have the proper number of characters.

2) Prices. It would not be difficult, in most cases, to define a range of values which would be accepted as valid. If a price outside this range was entered, the user would be asked to verify that figure (or correct it).

3) Dates. Unreasonable values for dates (such as the 32nd day of a month, or the year 1878) are easily caught. Once again, it is simply a matter of determining the minimum and maximum valid inputs.

Thus, you can see it's important to assume the user will occasionally make mistakes. But think about this – in *2001* (the book, not the movie), one of the boys with Mission Control commented, "We can make our systems fool-proof, but there's absolutely nothing we can do to protect them from sheer maliciousness."

Are the contents of your records potential targets for theft or tampering? They probably are. After all, what better way to cover up stockroom theft than to alter the inventory records. And what about payroll files and accounts receivable and . . . well, you know what I mean.

So far, of course, most computer-related crime has involved large financial institutions such as banks and insurance companies. But even though small businesses are a much less likely target, it's always wise to keep on your toes.

Conversion Time

Conversion time: That's the anxious period when the new system is being phased in and debugged. How long will yours take – a day? a week? six months? And to what extent will your operations be disrupted during this period?

In addition to breaking in the system itself, the phase-in period usually includes two more tasks: personnel training and record conversion. The first is self-explanatory. The second, as its label implies, is the process of converting your existing records into a format suitable for the computer. As you can see, both of these points are crucial for a successful conversion period.

Finally, don't forget to include your customers in your conversion plans. By this point, they should be aware of anything concerning the system that will directly affect them.

Don't be intimidated by all the suggested pre-purchase planning. I didn't intend to scare you away. On the contrary, the more thoughtful planning you do now, the more trouble-free your implementation will be, and the more rewarding your system will be in the future. □

Roulette on your PET with Bells and Whistles

BY DAVID M. CONLEY

Many times we hear about programs having "bells & whistles". This term indicates added extras in the program to make the output more interesting or useful. These extras are attractive features when writing programs for other people to use (especially children).

This article demonstrates added "extras" using some of the special features of Commodore's PET computer. A few features are unique to the PET, and for other systems you may have to work around them.

Now, I admit my program isn't a sophisticated simulation of a real roulette game. But you should find it fascinating when the computer spews out numbers at 15 to 20 spins a second. It'll sure teach you to keep away from straight number bets on the roulette wheel in Reno! In the end, you always lose.

However, the real point of all this is to demonstrate that even a simple thing like picking a number between 1 and 38 can be made interesting by adding extra bells and whistles.

Program Notes

The program (see Program Listing) is written in PET BASIC, which resembles other 8K BASIC systems.

We start with a simulated roulette wheel; the computer picks a number between 1 and 38, and then prints it out; 37 and 38 replace 0 and 00 on the real roulette wheel.

```
10 REM PET ROULETTE
80 X=INT (38*RND(TI)) +1   (X is the computer's number)
90 PRINT X
```

TI is what the PET's internal clock happens to read at that time. Other BASICs use 1 or 0. More about that TI later.

OK. We have our basic roulette wheel. Now what?

The following addition allows us to make a guess, then find out if we guessed right:

```
30 INPUT "YOUR GUESS": G:G=INT(G)      [G stands for Guess
   INT(G) keeps you from making G=3.456, for example]
40 IF G>36 OR G<1 GOTO 440        [592 or -5 won't work]
If you don't want a message, put in:
40 IF G>36 OR G<1 GO TO 30      [And don't put in lines 440 & 450]
100 IF X=G THEN PRINT "YOU WON!"
110 IF X<>G THEN PRINT "YOU LOST!"
180 END
440 PRINT "YOU CANNOT USE A NUMBER LARGER THAN 36"
450 PRINT "OR LESS THAN 0." : GOTO 30
```

Now we can add a way to bet on your number and add or subtract from your bankroll. We'll lay down a bet of $1 on each spin of the wheel.

```
50 INPUT "HOW MUCH MONEY DO YOU HAVE": M:M=INT(M):IF M 1
GO TO 50
[This line makes it impossible to start with less than $1, and rounds off entries that aren't whole numbers.]
120 IF X=G THEN M=M+35        [You win $35]
160 IF X<>G THEN M=M-1        [You lose your $1 bet]
180 GOTO 30                   [Ready for next guess]
```

So far, fine; but we can't tell how we're doing with our bankroll. So we add:

```
170 PRINT M
```

Turning the wheel one spin at a time gets tedious, so why not let the computer run as fast as it wants to, while we watch our money come and go? We add:

```
170 PRINT M;      [The semicolon saves space on the CRT]
180 GOTO 80
90                [Erasing that line]
100
110               [We don't need to see X for every turn]
```

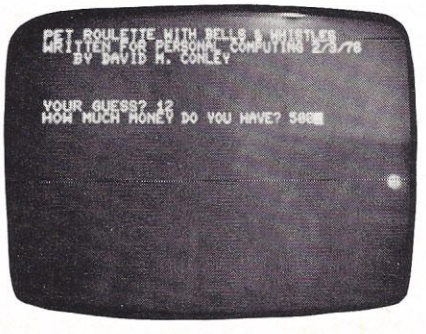
Now we're going like crazy. But the darned thing ain't gonna stop when we run out of money. So now we add:

```
100 IF M=0 THEN END
```

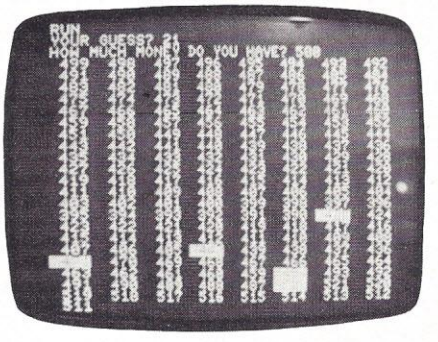
Great! Now it runs (and runs, and runs, and runs . . .) until we're out of money. But if you started with more than $500, you can play for quite a while.

How many times did we spin the wheel? We can add:

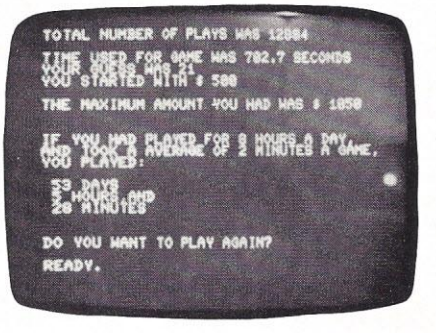
```
100 IF M=0 GOTO 190
110 T=T+1         [T stands for Times]
190 PRINT
200 PRINT "TOTAL NUMBER OF PLAYS WAS": T
210 END
```



The roulette-wheel program is loaded into the computer and we're all set to roll the ball around and make some money.



Like watching the ball rolling over the roulette numbers at Las Vegas, your computer gets dizzy displaying this array of numbers.



How long did the computer take to grab your dough? The PET measures time in "Jiffies". One Jiffy=1/60th of a second. We can make a stop watch for our game like this:

```
70 TI$="000000"      [Sets the timer to 0]
210 PRINT            [These extra PRINTs make it look nice]
220 PRINT "TIME USED FOR GAME WAS": (INT(TI/6)/10): "SECONDS"
[This line prints the time in seconds and 10ths of a second]
230 END
```

With all those confusing numbers filling your screen you've probably forgotten what you started with. So, we ask the computer to remind us:

```
230 PRINT "YOUR GUESS WAS": G
240 PRINT "YOU STARTED WITH $": M
250 END
```

You can see that this won't work! As the computer plays, M changes to 0! We need to add something else:

```
60 I=M               [I stands for Initial]
240 PRINT "YOU STARTED WITH $": I
```

Program Listing

```
10   REM PET ROULETTE
20   CLR
30   INPUT "YOUR GUESS": G:=INT(G)
40   IF G 36 OR G 1 GOTO 440
50   INPUT "HOW MUCH MONEY DO YOU HAVE" ;M: M=INT(M):
     IF M<1 GOTO 50
60   I=M
70   TI$="000000"
80   X=INT(38*RND(TI))+1
90   IF M>Y THEN Y=M
100  IF M=0 GOTO 190
110  T=T+1
120  IF X=G THEN M=M+35
130  IF X=G THEN PRINT "[RVS]": M:
140  IF X=G THEN PRINT "[RVS OFF]";
160  IF X<>G THEN M=M-1
170  IF X<>G THEN PRINT M:
180  GOTO 80
190  PRINT : PRINT"[CLR]"
200  PRINT "TOTAL NUMBER OF PLAYS WAS": T
210  PRINT
220  PRINT "TIME USED FOR GAME WAS": (INT(TI/6)/10): "SECONDS"
230  PRINT "YOUR GUESS WAS": G
240  PRINT "YOU STARTED WITH $": I
250  PRINT
260  PRINT "THE MAXIMUM AMOUNT YOU HAD WAS $": Y
270  PRINT : PRINT
280  PRINT "IF YOU HAD PLAYED FOR 8 HOURS A DAY,"
290  PRINT "AND TOOK AN AVERAGE OF 2 MINUTES A GAME, YOU PLAYED:"
300  PRINT
310  D=INT(T/240)
320  H=INT(T/30)-(D*8)
330  M=INT(T*2)-(H*60)-(D*480)
340  PRINT D:"DAYS,"
350  PRINT H;"HOURS, AND "
360  PRINT M;"MINUTES."
370  PRINT : PRINT : PRINT "DO YOU WANT TO PLAY GAIN?"
380  GET A$ : IF A$="N" THEN END
390  IF A$="Y" GOTO 410
400  GOTO 380
410  PRINT "[CLR]": PRINT : PRINT "          **** NEW GAME ****"
420  PRINT
430  GOTO 20
440  PRINT "YOU CANNOT USE A NUMBER LARGER THAN 36"
450  PRINT "OR LESS THAN 0." : GOTO 30
```

(No, I don't know what happened to line 150! Don't worry about it. Everybody loses something.)

An interesting thing to know now is how much money you won before you went broke. It's only a small consolation because there's no doubt you will eventually go broke. Most people do. (In this instance, the computer's roulette is a good simulation of the real thing.) So we add:

```
90 IF M>Y THEN Y=M      [Y stands for Y Not?]
250 PRINT
260 PRINT "THE MAXIMUM AMOUNT YOU HAD WAS $": Y
270 END
```

We've got what we're looking for! What else can we add? If we were playing for real, how long was the game that we had just played? With a bit (Ha!) of calculation we come up with:

```
270 PRINT : PRINT
280 PRINT "IF YOU HAD PLAYED FOR 8 HOURS A DAY,"
290 PRINT "AND TOOK AN AVERAGE OF 2 MINUTES A GAME,
    YOU PLAYED:"
300 PRINT
310 D=INT(T/240)
320 H=INT(T/30)-(D*8)
330 M=INT(T*2)-(H*60)-(D*480)      [M is now Minutes]
340 PRINT D: "DAYS"
350 PRINT H: "HOURS, AND"
360 PRINT M: "MINUTES."
370 END
```

Since we want the output to appear nice and tidy on the CRT, we add:

```
190 PRINT : PRINT "[CLR]"
```

To clear the screen on the PET, press the CLR key between the quote marks. The symbol that appears is a reversed heart symbol. This looks a little weird, but it means "Clear Screen and Return Cursor to the UL corner".

Now we can make it easy to start another game by using the PET's GET function:

```
370 PRINT : PRINT : PRINT "DO YOU WANT TO PLAY AGAIN?"
380 GET A$: IF A$="N" THEN END
390 IF A$="Y" GOTO 410
400 GOTO 380      [The computer goes around and around and . . . ]
410 PRINT "[CLR]": PRINT "          **** NEW GAME **** "
420 PRINT
430 GOTO 20
```

And to clear up all those variables you've stored, add:

```
20 CLR     [This doesn't clear the screen: it sets all variables to 0]
```

If you don't do that, then part of your last game will get all mixed up with the new game. Besides, line 430 has to have a line 20 to go to.

For the last thing, wouldn't it be nice to see where you won an occasional spin among all those losing numbers? With the PET's REV feature, when you win, you can make the next listing of your bankroll appear as black numbers on a white background. Losses will appear normally. Add:

```
130 IF X=G THEN PRINT "[RVS]": M:
140 IF X=G THEN PRINT "[RVS OFF]";
170 IF X<>G THEN PRINT M:
```

Now type LIST, return, and hold down the REV key. The whole thing should look like Program Listing and should scroll slowly enough to read as it goes. □

THUNDER

BY RICHARD J . LUTZ

"Yes, sir, may I help?" Direct, efficient, unattractive, he thought. Even antiseptic. No small talk here.

"Hello. Roger Scott is expecting me. David Bowman."

Committed now, he thought. Even before the few words were out, the digitized voiceprint would be recognized at Center, and Center would be noting the presence of David Bowman in the offices of Haskell-Forbes, Scott, Pierce. Even more important, Center would know Bowman was there for the first time. Immediately, memory would be set aside and the situation analyzed.

"Mr. Scott is on schedule. Please wait." The eyes were suggesting a chrome- and flexhyde couch, and he turned toward it as she reached to still a telephone's purring.

David Bowman reviewed the approach he'd planned. It would not be easy. Truth would not be enough. Persuasion would be important. Being a good salesman would be less than the occasion demanded. More important to convince – to sway Scott – to make Scott an ally.

Age 33, thought Bowman. Here I am at age 33 running around covering for a goddam superkid with more brains than sense, more chutzpah than smarts. But this would be the end of it.

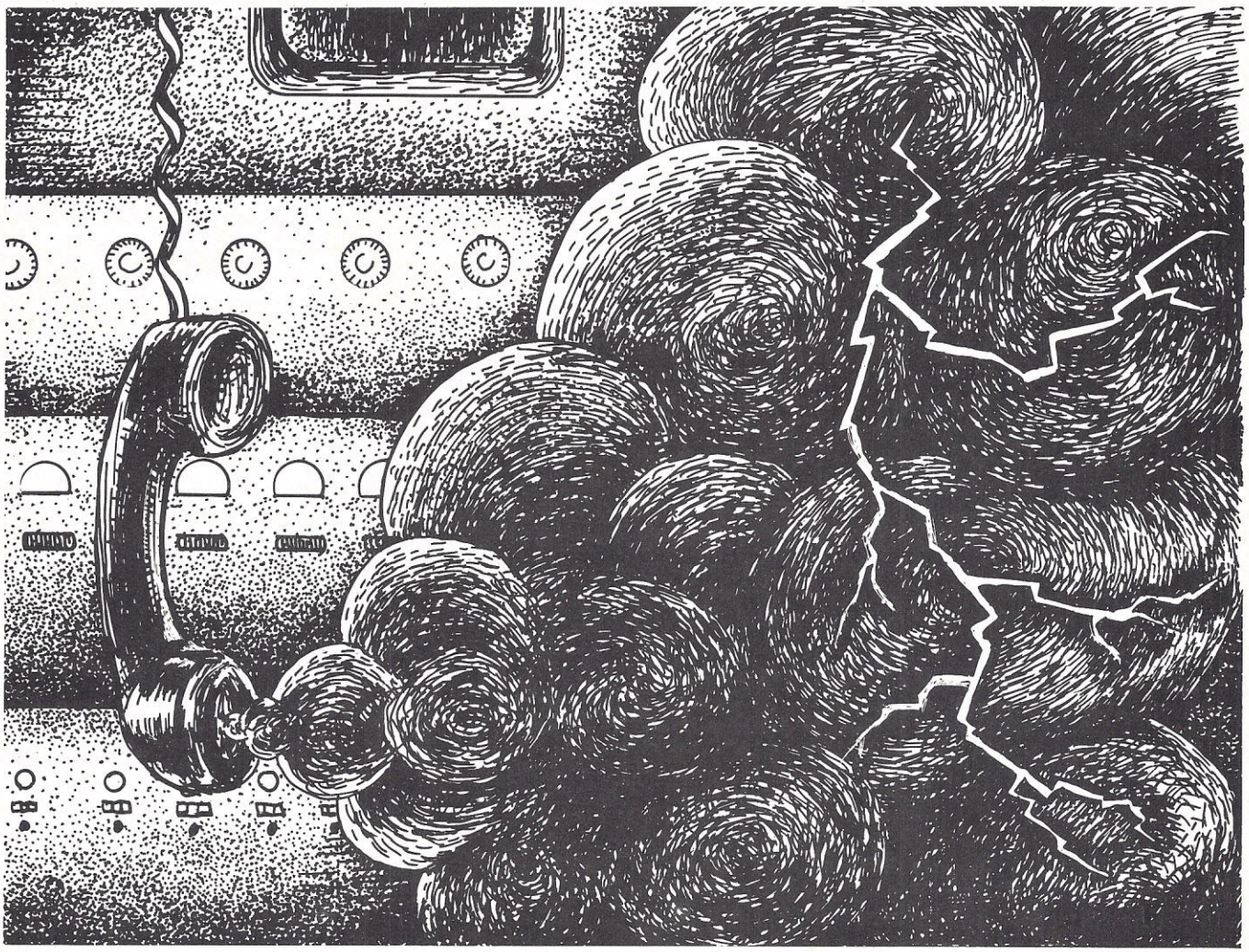


Illustration by Barbara Leonard

No more calls. No more phone. The kid could be isolated.

Bowman shook himself mentally. Enough of anger with the kid. Better to concentrate on the line of argument. Scott would have to see the futility of pursuing the matter any further . . .

The voice, just mildly annoyed, came through his fog of deliberation. She must be repeating, perhaps even for the third time. "Mr. Bowman. Sir."

"Yes . . . I'm sorry . . . I was think – "

"Mr. Scott is free." Again the eyes, indicating the door. Perhaps it wasn't Scott's style to greet visitors in the reception area, but more likely he was going to make this as tough as possible.

The *thuck* from the doorframe told him she was releasing a latch from her desk.

Scott stood to greet him, at least. Younger than Bowman expected – not the legal elder statesman in the telephone impression. Smiling just slightly. Curiosity creeping into the lines around the eyes. But no words – just the outstretched hand.

"Thank you for this appointment, Mr. Scott. On short notice, I mean."

"Not at all. Please sit. This is an important matter, and time is getting short." Scott sat behind his desk. "My client is most eager to see this matter resolved, as you must imagine."

"Naturally."

"Mr. Bowman, since you represent Thunder and I represent Stadium Sound, I think it appropriate we record our conversation, if you don't mind – to protect all the interests involved here." With only slight hesitation, he was reaching for controls.

"No," said Bowman, immediately realizing he sounded more firm than he intended. "I'd prefer we chat first, if you don't object. This is a difficult situation." To himself, Bowman cursed the bead of sweat that appeared on the bridge of his nose.

"As you say." Scott agreed too easily. Must be some recording device running already. Damn. Nothing to do about that, except be reasonably confident no tapes or digicubes would show up in court, at least. For that, Scott would need Bowman's voiceprintable assent to the recording on a digicube sealed against editing. Still, having the session on any kind of record troubled Bowman.

"Look, later if it still seems important to you, I'll record some kind of statement for you. OK?"

"Surely. Would you mind filling me in on your connection with Thunder? Are you counsel?"

"No, no. I'm no attorney. Thunder isn't represented by anybody. It hasn't been necessary and I don't anticipate it will be. I hope it won't be, I mean." Oops, too much. Wrong move to tell a lawyer that the hottest new rock group in the country didn't need legal advice.

Scott put on his professional "impatient with laymen trying to play attorney" face. "I don't think it would be fair for us to proceed without suggesting to you that this situation could very well lead to a courtroom and some sort of judgement. To go on without legal advice could be a grievous error."

Spoken for the recorders, thought Bowman. Boilerplate legalese to get Scott off the hook later if accused of trapping Bowman into admissions without some legal protection. Officer-of-the-court and all that. No matter.

"I see your point, Mr. Scott. I believe you'll see why legal counsel shouldn't be required in this matter.

Bowman cursed the bead of sweat that appeared on the bridge of his nose.

"Just what is your connection with Thunder? Manager?"

"In a way. I think I can make all that clear."

"All right. Let me just say, first, that as counsel for Stadium Sound Enterprises I cannot conceive of your client. . . uh. . . of the group pulling out. We feel we have a firm contract for the appearance. So far, we've heard no reason for your statement there will be no concert." Scott leaned forward. "In fact, Mr. Bowman, even after you called, we had further confirmation Thunder would appear."

"Oh, no. Good God, how?"

"Just a phone call. Rashin, the same man we dealt with originally."

"Oh, boy."

"Well, in any case, perhaps I should let you go ahead and perhaps we can sort this out between us."

"Mr. Scott, there is no Rashin. He doesn't exist." Damn, thought David Bowman. This wasn't the way he'd intended to get into this thing.

"Nonsense," declared the attorney. Now it was really out the window. "Mr. Bowman, are you aware that Mr. Rashin executed a dataline contract for the appearance of Thunder, and we have it in electrofile here? If you would like to see it. . ."

"No. I know that. I know about the contract. Let me explain about Rashin. He's a creation – pure imagination."

"No, no. Rashin checks out. We didn't offer an advance, or pay an advance to his account for that matter, without determining he's legitimate. What in hell do you mean 'pure imagination'?"

Got to get this back on the planned track, thought Bowman. "Mr. Scott, forgive me. If you'll let me start from the beginning, I'm sure you'll . . ."

"OK. Start somewhere. Please start somewhere."

"How old are you, Mr. Scott?"

"How old? I don't know what relevance. . . Hell, alright, I'm 46."

"Well, I'm 33. The relevance of your age is that when you and I were kids – and this is true of anybody over 30 these days – when we were kids the most sophisticated gadget around was a pocket calculator or a programmable microwave oven. At least in common use."

"Yes, true." Scott was cooling down a bit.

"That's all changed now."

"Yes."

"I mean, who could have foreseen the kinds of things our world has today. When my father started having heart problems they implanted a new experimental telemedic and told him not to worry unless he had to punch his override in some emergency. Weird."

"Weird? Certainly extraordinary. Pierce has one. Oldest of the partners. Something I'll never understand. I don't mean the benefits. I mean how it works."

Good. Back on track now. This might go well after all.

"Well, I've had telemedic explained to me." Bowman hurried to get in his point before Scott would see it as a digression. "They code the telemetry from the body function sensors so the transmissions identify the specific individual involved. The readout goes to Center on a time-share dataline and the body's parameters are

checked against medical history and the diagnostics algorithm. . ."

"You're a physician," Scott declared, interrupting the flood of jargon with a wave of his hand.

Still trying to hang a profession on me, thought Bowman. "No, I'm not a doctor. I've just had it explained. Look, believe me, this is relevant – at least peripherally. Let me go on."

"Go on."

"Anyway, the patient gets a warning if Center computes that system parameters are going to be violated, that is, if the person with the telemedic is overstresssing in some way."

"I don't follow this too well, I'm afraid."

"OK, I guess I do tend to slip into technical language a bit. Anyway, they're working on even more advanced models where the telemedic simulates nerve impulses when signalled by Center, and telemedic tells the guy's brain to slow down, to rest the subject automatically. Even without the subject knowing, at least most of the time. My father's got one of these new ones. He's got an override button under the skin just below his ribcage, in case Center starts phasing him down during a real emergency. Aside from some real crisis, telemedic just cues his brain to feel sleepy in the middle of tennis, or whatever. There have been a few embarrassing moments, but they're nothing compared to an all-out coronary."

The bemused expression on Scott's face told Bowman he was going too far afield.

"Anyway, Mr. Scott, the point is that you and I had no clue these kinds of things would be coming along, and they still mystify us."

"Me more than you, obviously."

"Frankly, I've had to get interested in this stuff, sort of in self defense."

Another quizzical look. "Mr. Bowman, do I sense correctly this is going to take some time?"

"Yes, I guess you're right, but. . ."

"No problem. Just let me arrange a couple of things." Scott punched a button on his desk and a gas plasma display read out his appointments. From his seat David Bowman noticed one item flashing – usually an indication it's overdue. Scott reached for a bottle of pills with one hand and punched an intercom button with the other.

"Stacey, do you have my readout up?"

"Yes, sir." The voice came back from a speaker in the desk.

"Would you delete the medical? I'm taking it now." He swallowed a pill without water. The flashing stopped. "Thank you. Now would you see if Haskell or one of his boys can take the Shriver desposition when she comes in, and check on whether we can move Roberts to tomorrow's list. You don't have to get back to me unless some of that won't work out."

"Yes, sir."

Scott looked up at Bowman, a signal to continue.

Bowman did. "Even that schedule-dex or whatever you call it. We didn't know that would be so widespread, did we?"

"The point you are trying to make, I gather, is that the world is full of electronic wonders."

"When we were kids the most sophisticated gadget around was a pocket calculator or a programmable microwave oven. That's all changed now."

"Computer-based wonders, right. The point is, Rashin is not real. He's . . . it's . . . electronic. The whole business. He's even got an identity that's been insinuated into Center's records so thoroughly that Center – the Center computer, that is – thinks he's real. Medical history, purchasing records, independent income, all that. There's even a digital voiceprint on him, and Center thinks he has coffee everyday in a little diner near my home, and. . . oh, there's lots of other stuff."

"I spoke with him when we executed the contract."

"Of course. He. . . *It* has a voice, as I said. It's not just a digital simulation stuck onto the diner's dataline to Center. It *is* real, more or less, you can carry on a conversation and think you're talking. . ."

"Are you telling me I've been dealing with a robot or something?"

"No, not quite. The fact is you were talking with a kid – a 12-year-old."

"Who disguised his voice, I suppose. Come on, Bowman, how gullible. . ."

"That's right. Disguised it. Not so much disguised it as processed it. Electronically. He talks into this processor in his normal voice, and out comes, well, Rashin's voice."

"A kid? That's hard to swallow, I must say."

"Well, it's true. Kid's a genius. Hard to restrain him, frankly."

"Who the hell *is* this kid, if he really exists?"

"Look, I can understand your doubting me. Just bear with me, please. Excuse me for being blunt, but is this conversation being taped, or digitized, or anything?"

"You said I could not record. . ."

"But you are anyway. That's OK. I understand. I know you have to protect yourself and keep records and all that. I don't mind if you're recording. I just need to know if it's being fed to Center."

The question startled Scott. He leaned back in his comfortable chair. Bowman realized he'd stunned Scott by knowing that Center could be tied in. Silence. Bowman decided to try again.

"I told you. I've had to learn about some of this stuff. Self-defense. I asked you about it, not to call you a liar, but because this conversation *really* needs to be off the record, at least from Center. If you want to record it, OK, but if you're feeding Center online I beg you, please break the link. If you really think you have to supply Center on it, you can send the digital recording through later on, OK? I'm sure you'll see why you must not do that."

Scott just nodded, and reached for a switch on his desk. Bowman knew he could be faking it, but had to take the chance. No choice. Quickly, he reviewed what he'd said about Rashin. OK, so Center would know now that Rashin was no entity. Created by a kid, 12 going on 55. That much couldn't be helped now.

"Thank you," Bowman said, glancing at the switch Scott had touched. "You asked who the kid is. I'll tell you. He's my son. He's a goddamned

genius. Sooner or later, Center will find out about him; I know that. I just want him to have a few more years."

"I see. Let's suppose I buy the story Rashin doesn't exist, that he's really a pre-pubescent electronic wizard who represents the country's best-selling rock group. And that he executed a contract with my client for a Thunder appearance. Now you'll have to admit that's just going to fall short of believability, but let's assume I buy it. Are you trying to convince me the contract is invalid because the executor is under age? Is that it? And you want me just to take it at that, and let Thunder off the hook for this appearance? That's how I'm to represent Stadium Sound in this matter?

"Mr. Bowman, isn't it possible Thunder has rocketed to success far faster than you and Mr. Rashin thought possible? And that you want to boost the price for this appearance? Frankly, Stadium Sound is pleased it saw this success coming and booked the group early. At a decent, uninflated price.

"And speaking of Stadium Sound what the hell do I tell them if I buy your story? That they've been doing business with a group that's got a mischieveous robot for an agent? What, for God's sake?"

"I wish it could be that simple," said David Bowman, feeling more beads of sweat between his eyes.

"There's more?"

"There's more."

"Oh, God. All right, go on. I'm sorry to get heated up on this, Mr. Bowman. Even if I wanted desperately to believe you, this is all a little far-fetched, don't you think? Would you like a drink?"

"Yes, please." Bowman didn't want a drink, but the interruption would give him a moment to think.

"Scotch?"

"Gin. Anything with gin, if you have it. Tonic, perhaps."

Scott went about the routine of mixing drinks at an old-fashioned bar. There's one thing, thought David Bowman, that hasn't gone all the way to wires and microchips, not yet anyway. Men still could take pride in their mixology, even though autotenders were getting good.

Scott was taking Scotch, neat. Just a little. Prudent drinker, thought Bowman.

"Far fetched I admit," he said.

"And I have to admit that without some sort of proof it's pretty hard to go to your client and explain why you cannot follow through with a lawsuit or some action against Thunder."

"Well, we'd probably be forced to bring suit against you, since presumably you're really the group's agent, and not your son. And unless you're now prepared to deliver the group for an appearance . . . "

"I'm not. Please let me continue."

"Continue."

"When you buy records, you know how the cashier runs that light pen over the black-and-white stripes on the jacket? And the register records the sale and rings up your bill?"

"Yes."

"Good. Well, those stripes also note what the record is, for inventory purposes. The information goes onto a dataline and goes through to a warehouse, and it's used to figure out how fast a given item is selling – a disk, a tape a digimuse or whatever the form of recorded music you're buying. The warehouse uses the information to keep up with sales, keep items in stock, and to inform the manufacturers so they can gear their production to demand, and so forth."

"Right."

"That's what's behind the marketing revolution we've been seeing in every field in this country. Instant corporate response to public preferences expressed through buying decisions. Really remarkable, when you think about it."

"Right." Bowman saw Scott's attention wandering. Back to the point.

"This situation could well lead to a courtroom and some sort of judgement. To go on without legal advice could be a grievous error."

"A couple of years ago, Ricky – that's my son's name – tapped into a line by mistake and became fascinated with the data and what it meant. Spent night after night trying to decode it, make sense of it. Like a hobby, but more an obsession. He's got this computer kit and damned near a roomful of gadgets. Anyway, he finally did decode it, and it turned out to be master inventory information for a record company – and through that he was able to ply into the company's whole data system."

"Criminal act, you know. Interception."

"Of course. That's one reason I wanted to be sure Center was offline."

"Very well. Go on."

"Ricky has a hell of an imagination, and frankly I just didn't, and don't, know enough about this stuff to know what he's up to from one day to the next. He bought the best-selling records on the market. Spent weeks running them through his system and did something he calls 'digital appeal analysis,' whatever the hell that is. Has something to do with finding out why particular records sell well and others don't."

"Does he play some instrument?"

"No. . . and yes."

"Is he one of the group on the Thunder album cover?"

"Yes. . . and no."

Scott laughed out loud. Bowman laughed at himself, but nervously. Trying not to sound evasive, he was sounding evasive as hell.

"You'll understand in a minute. You see, after Ricky got tired of his appeal analysis project, he fooled around with patching his computer into other people's computers. Apparently, he happened onto a really big one, by chance. We still don't know where, or whose it is, but it's probably a big government installation because there's a lot of spare memory space and a lot of idle time, he says."

"That's also. . ."

"Also an offense. I know. Ricky somehow fed all his analysis of music into this bigger computer, and then got the bigger computer to work with him on coming up with rules for the music – what seems to sell best, and how it's structured, and what the sounds are like."

"Ingenious."

"You're telling me."

"Diabolical, maybe."

"Maybe. Well, to make a very long story a little shorter, and perhaps you

can see where I'm leading, the kid found a way to get the big computer to actually compose the music for him, even lyrics, and then, believe it or not, to prepare a master recording digitally."

"*That's* Thunder?"

"That's Thunder."

"The biggest thing in music is something put together by a kid and a computer?"

"Right."

"The singers?"

"The works."

"Oh, my God."

"You're tellin' me oh-my-God. Thunder is like Rashin, more or less. Thunder doesn't exist, except as a program in some big government computer somewhere. And I don't even think the government knows. If they did I don't see how Ricky could have ever finished the album."

"What does it do, feed it back to him on the phone?"

"In a sense. The music is pretty complicated, but it's all digitized and fed back slowly. It takes a week to get all the tracks laid down, and you can't really play it until it's all recorded, a little bit at a time. Even then, you know, Ricky fiddles with it – puts in a wrong note somewhere, maybe drops a rhythm pick-up – makes it more credible."

"What about the group in the picture on the album cover?"

"Magma Records uses a digital process to work up their color printing. Ricky's into that part of their computer, too. It's all digitized art, with a little help from Ricky. Hell, even the write-up on the cover is an assembly of typical stuff from other album covers, from the successful ones, with names changed and so forth."

"Why would Magma records agree to . . ."

"Hold it. Before I go on with this, do you believe me? Because if you don't, it's crazy to go on."

"Mr. Bowman, this is a marvelous story, and I'm fascinated, I'll admit. But as a basis for a legal decision, I doubt I can represent my client by turning fascination into advice."

"But *you* believe me?"

"Just as a basis for continuing this discussion, I do. Tentatively, I mean."

"OK. Well, the long and the short of the album being released – since you're asking why Magma would do it – is that Ricky ran the whole operation right through their computer. He ordered up the production, fed the digitized music over data lines on weekends, had the jackets printed up, adapted some of their press releases on other groups and albums and budgeted in a promotion campaign for Thunder.

"He's even got it set up, Mr. Scott, so Magma's electrofiles contain a contract with Thunder, and royalties and other payments go automatically to a bank account Ricky injected into the same bank where Magma does its business – except it's all payable to Rashin.

"Frankly, I'm getting worried about income-tax time, but Ricky says it'll all work out."

"Are you suggesting Magma executives don't know anything about all this?"

Oh, they know. Ricky's getting a hell of a boot out of the communications going through the computer. They're mystified, but why the hell should they try to stop it? Magma stock is really taking off with Thunder under contract to them.

"Even will all the trouble the kid is causing me with this, I have to admit I admire him. The whole thing is slick as a pig's oink.

"There have been a few embarrassing moments, but they're nothing compared to an all-out coronary."

"But it's driving me nuts. Ricky's been playing Rashin on the phone, like with you, but it's getting out of the computer realm and now I'm running around covering for him. It's too much."

"You'll probably need an attorney, as much as you want to avoid it."

"But the thing that worries me most is what will happen if Center gets wise. So far it's been play, albeit a little overcommitted to computer play. But if Center gets hold of him, you can damned well figure they're going to have him working on a goddam bomb or something."

"Yes. I understand your worry."

The two men sat quietly, pondering the consequences and realizing they couldn't foresee half of them. Finally, Bowman shook himself back to his present problem.

"Now, Mr. Scott, as to how we will explain all this to your client..."

"Hold it on the 'we.' I think this is probably too good a story for you to have made up, but I couldn't get Stadium Sound to back off without some proof."

"So I'll prove it to you."

"How? How do you propose to prove it? But wait; suppose you do. There's still the advance against the gate for the concert. What about that?"

"Yes, the advance. Gone, I'm afraid. Stadium Sound will just have to write it off. As soon as it was credited in, Ricky credited it out for a TDP45-11. They delivered it last week. Had to sell the piano to make room for it. Look, the advance is just money. Stadium Sound can make it up in one show. I need your promise that my son's *life* won't be destroyed."

"What about the proof?"

"Call Rashin. Call him now. Put him on your speakerphone. Engage him in a conversation about the booking. I'll do the rest."

Scott fingered his switches again.

"Yes, sir."

"Get me that Rashin that represents Thunder."

"Yes, sir."

Finally, David Bowman felt free to wipe away the accumulation of beaded sweat between his eyes.

"Rashin here." The deep voice from the phone sounded like a TV stereotype of a showbiz agent.

"Mr. Rashin, Roger Scott here. We need to arrange a brief meeting with Thunder and you to discuss arrangements, security and the like."

"A meeting will be quite difficult, Scott." The voice was convincing as hell, Bowman thought, hearing it in this law-office setting. "Thunder is in seclusion, preparing. They meditate and rehearse and have a very specific routine. Can't be disturbed."

"But this will be their first public appearance. There have to be arrangements. Are they stage ready?"

"They are stage ready, believe me. Their album speaks for itself, does it not?"

Bowman stood, and walked to the desk. The kid was taking this much too far. This wasn't computer play or evasion; the kid was outright lying.

"Mr. Scott. I repeat." Insistent now. "The album speaks for itself, does it not?"

Bowman spoke: "Can it, Ricky. The jig's up."

From the speakerphone came the child's voice: "Aw, Dad, whadja go and do that for?" □

Petals Around the Rose Revisited

BY KEN JACKMAN

Back in our September/October 1977 issue, we introduced a simple dice game called "Petals Around the Rose" We explained how you play the game then, but we refrained from printing the accompanying program since it would give the game away.

But by now, you've either figured out the algorithm or given up trying. So we're going to give you Kenneth Jackman's program for Petals Around the Rose – but not the algorithm. It's not obvious from the Program Listing, so we're going to let you figure it out for yourself.

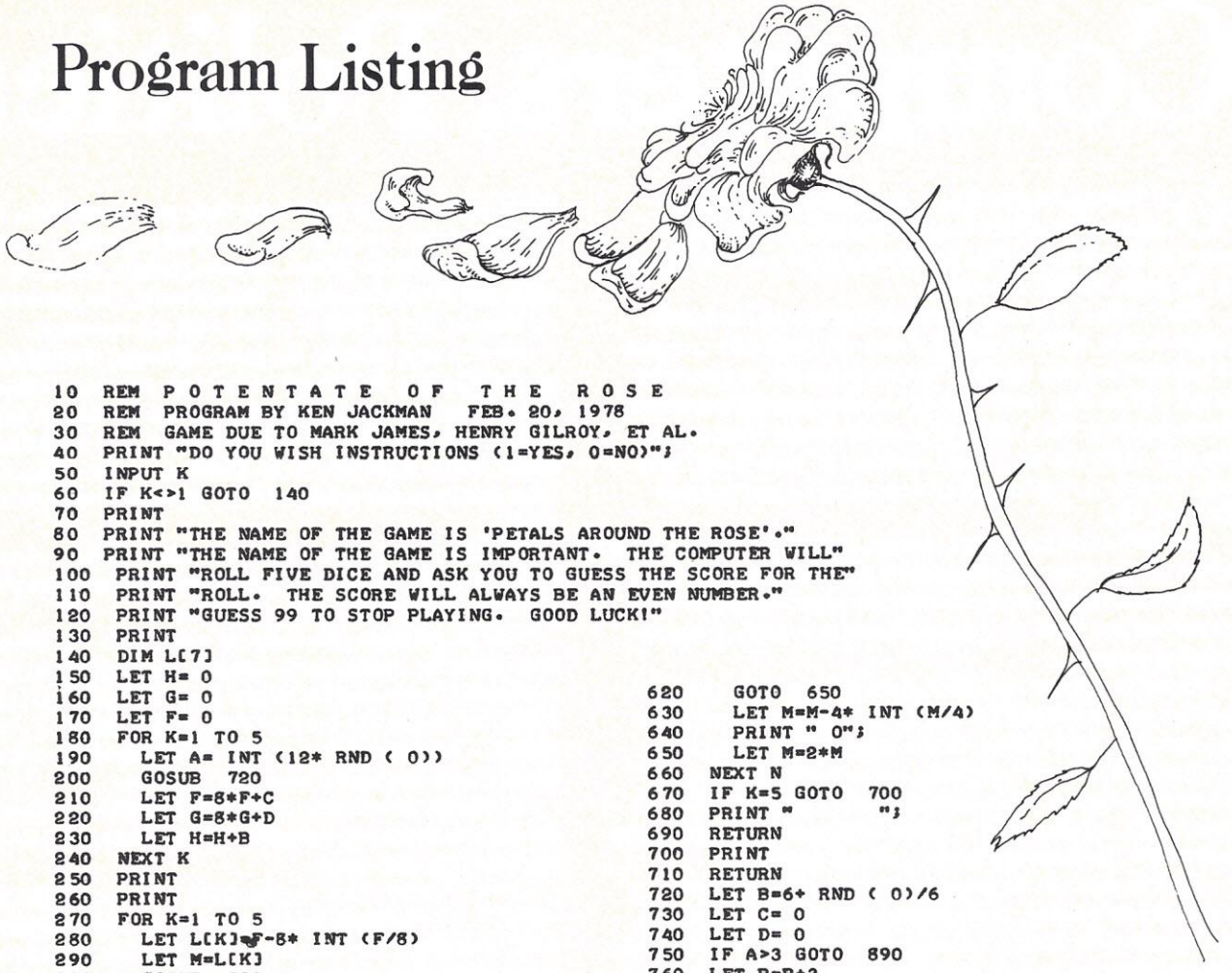
We will repeat the rules for those of you who missed the article. Here's how you play the game: The Potentate of the Rose, who knows The Secret, can tell newcomers only three things: 1) The name of the game is Petals Around the Rose, and the name is significant 2) The answer to any roll of dice is always even; and 3) He can tell you the answer to any particular roll of the dice.

The Potentate then rolls five dice, and tells you the answer. Rolls continue until a newcomer thinks he knows The Secret (the algorithm generating the answers). If the newcomer guesses six successive answers correctly, the Potentate assumes he has indeed discovered the algorithm, and dubs the newcomer a Potentate of the Rose in his own right, charging the new Potentate with the duty of spreading the game around. We think you'll find "Petals" infectious. Have fun!

Sample Run

```
DO YOU WISH INSTRUCTIONS (1=YES, 0=NO)? 0

0  0        0    0 0 0    0  0    0  0
0  0                      0
0  0    0        0 0 0    0  0    0  0

GUESS THE SCORE? 12 NO, IT'S 4

0  0    0  0    0  0       0
   0       0       0             0
0  0    0  0    0  0    0

GUESS THE SCORE? 4 NO, IT'S 12

0  0    0 0 0    0 0 0          0  0
0  0                      0
0  0    0 0 0    0 0 0          0  0

GUESS THE SCORE? 16 NO, IT'S 0

0       0       0  0             0
   0                          0
      0       0    0  0

GUESS THE SCORE? 7 NO, IT'S 2  (THE SCORE IS
ALWAYS EVEN)

0  0       0          0     0
0  0    0       0          0        0
0  0    0          0             0

GUESS THE SCORE? 10 NO, IT'S 6

0 0 0    0  0                   0     0
            0          0
0 0 0    0  0                0        0

GUESS THE SCORE? 4 ..YES..
```

Program Listing



```
10  REM  P O T E N T A T E   O F   T H E   R O S E
20  REM  PROGRAM BY KEN JACKMAN   FEB. 20, 1978
30  REM  GAME DUE TO MARK JAMES, HENRY GILROY, ET AL.
40  PRINT "DO YOU WISH INSTRUCTIONS (1=YES, 0=NO)";
50  INPUT K
60  IF K<>1 GOTO  140
70  PRINT
80  PRINT "THE NAME OF THE GAME IS 'PETALS AROUND THE ROSE'."
90  PRINT "THE NAME OF THE GAME IS IMPORTANT.  THE COMPUTER WILL"
100  PRINT "ROLL FIVE DICE AND ASK YOU TO GUESS THE SCORE FOR THE"
110  PRINT "ROLL.  THE SCORE WILL ALWAYS BE AN EVEN NUMBER."
120  PRINT "GUESS 99 TO STOP PLAYING.  GOOD LUCK!"
130  PRINT
140  DIM L(7)
150  LET H= 0
160  LET G= 0
170  LET F= 0
180  FOR K=1 TO 5
190    LET A= INT (12* RND ( 0))
200    GOSUB  720
210    LET F=8*F+C
220    LET G=8*G+D
230    LET H=H+B
240  NEXT K
250  PRINT
260  PRINT
270  FOR K=1 TO 5
280    LET L(K)=F-8* INT (F/8)
290    LET M=L(K)
300    GOSUB  590
310    LET F= INT (F/8)
320  NEXT K
330  FOR K=1 TO 5
340    LET M=G-8* INT (G/8)
350    GOSUB  590
360    LET G= INT (G/8)
370  NEXT K
380  LET H=H-12* INT (H/12)
390  FOR K=1 TO 5
400    IF (L(K)-1)*(L(K)-4)<> 0 GOTO  420
410    LET L(K)=5-L(K)
420    LET M=L(K)
430    GOSUB  590
440  NEXT K
450  PRINT
460  PRINT
470  PRINT "GUESS THE SCORE";
480  INPUT K
490  IF K=99 GOTO  1110
500  LET H=2* INT (H)
510  IF K=H GOTO  570
520  PRINT " NO, IT'S";H;
530  IF K-2* INT (K/2)= 0 GOTO  550
540  PRINT " (THE SCORE IS ALWAYS EVEN)";
550  PRINT
560  GOTO  150
570  PRINT " ..YES.."
580  GOTO  150
590  FOR N= 0 TO 2
600    IF M>3 GOTO  630
610    PRINT "  ";
620    GOTO  650
630    LET M=M-4* INT (M/4)
640    PRINT " 0";
650    LET M=2*M
660  NEXT N
670  IF K=5 GOTO  700
680  PRINT "       ";
690  RETURN
700  PRINT
710  RETURN
720  LET B=6+ RND ( 0)/6
730  LET C= 0
740  LET D= 0
750  IF A>3 GOTO  890
760  LET B=B+3
770  IF A>1 GOTO  810
780  LET D=D+2
790  LET B=B+3
800  RETURN
810  LET C=C+1
820  LET B=B+1
830  IF A>2 GOTO  780
840  LET B=B+11
850  GOTO  790
860  LET C=C+1
870  IF A>2 GOTO  790
880  GOTO  780
890  LET C=C+4
900  LET B=B+2
910  IF A>5 GOTO  970
920  LET B=B+2
930  IF A<5 GOTO  950
940  GOTO  780
950  LET B=B+11
960  GOTO  790
970  LET C=C+1
980  LET B=B+8
990  IF A>6 GOTO  1030
1000  LET D=D+3
1010  LET B=B+5
1020  GOTO  780
1030  IF A>7 GOTO  1070
1040  LET B=B+5
1050  LET C=C+1
1060  GOTO  860
1070  LET B=B+7
1080  IF A<10 GOTO  780
1090  LET B=B+10
1100  GOTO  790
1110  PRINT "  THE SCORE WAS";2* INT (H)
1120  STOP
```

Computers, Politics

On Tuesday nights Stuart Appel works as a county planner for APEX County. In reality he is a landscape architecture student pretending to be a county planner in the game of METRO-APEX.

While APEX is the hypothetical urban area for which Appel works, METRO-APEX is also the name of a computer simulation game he plays to learn what the professional world will be like. Each Tuesday landscape architecture students gather at the College of Environmental Science and Forestry (ESF) at the State University of New York, Syracuse, to play APEX as part of a required course in Land Planning Use.

The School of Landscape Architecture at ESF uses METRO-APEX to familiarize students with real-life situations they will encounter as future professionals. So what appears to be a classroom in a state of mass confusion is actually a creative learning process.

Played at universities throughout the country, this training tool was developed jointly by the Universities of Michigan and Southern California at a cost of $1 million. It has been used at Cornell University for about 6 years in a government course to teach environmental and political issues. Now APEX is being used to teach landscape architecture students their relationships to the political process.

The key element of the game is interaction, achieved by students role playing, or simulating real-world experiences. By assuming the basic roles of industrialists, developers, politicians, planners and air pollution control officers, students are offered several advantages over traditional classroom lectures. By gameplaying, students interact, gaining self-confidence and insight into how their professional counterparts function. Without harming a community, they can experiment with ideas and approaches, and gain an understanding of people who harbor a different point of view.

"Because our students see through the eyes of designers and planners," say ESF Professor Allen R. Lewis, "they tend to view issues as pro-environment and things political as tainted. By changing these naive black and white attitudes, our students will be better able to deal with reality as professionals."

The game comes as close to reality as possible. In fact, students often become so involved in their roles that they are carried beyond the classroom. One student may stop another game participant on Thursday to argue an unsettled point from Tuesday night.

At the beginning of the semester in September, 92 students were assigned roles from their preference lists. Besides the basic roles, students were selected to portray representatives from pressure groups, lawyers, judges, members of the press and broadcast media. Theoretically, the game could go on forever. But at ESF it ended with the semesters.

No one wins or loses the game, although it involves winning and losing. Politicians win or lose elections. Litigants win or lose suits. And industrialists and developers earn financial profits or losses. There are no 1, 2, 3 rules like most games. Instead, players make decisions on the basis of their roles and learn the results of their actions by computer feedback.

To role play a county planner thrills student Appel because that's his career goal, and what he does while playing METRO-APEX will enhance his future.

Like all students first assigned a role, this fourth-year student reads the master game manual and a computer printout outlining his role and his record during the previous year. Initially, he studies the metropolitan area of APEX County consisting of a central city, a suburb, two townships, a business district, elite residential areas and a ghetto.

All the problems and tensions of today's urban areas are programmed into the computer. It is against this setting that students play their roles.

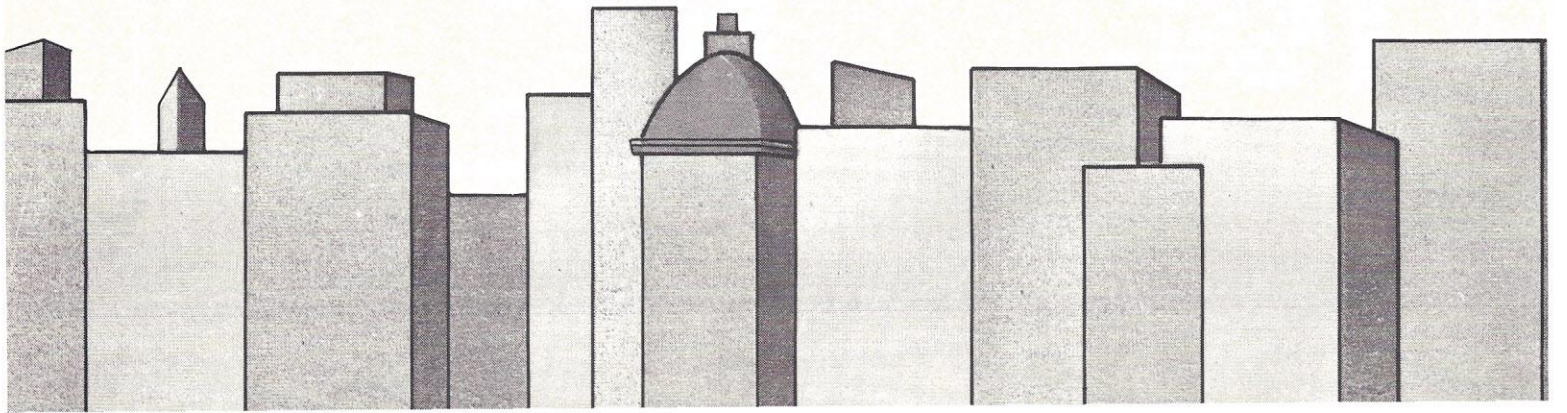
Each Tuesday night, Appel grabs a copy of "METRO-APEX", the computer and student generated newspaper. After gleaning it for data he can use to make decisions, he meets with other students to watch a 15-minute news broadcast, prepared by students playing the media roles.

Each student fills out weekly worksheets representing a complete year of policy decisions for their respective roles. They must also vote on issues.

All actions that take place during each gaming session are coded and fed into the computer, which generates new printouts for the start of the next 2-week cycle (one year's work for a professional).

Groups meet throughout the classroom. Appel moves to an area marked "County Planning Board", where he meets his five associates – students by day, planners on Tuesday nights. This group just met over the weekend to research a 25-year development plan for their county. While extra meetings are optional, additional research means students can play their roles more effectively.

One planning board member repeatedly failed to make the extra meetings. The group chastized him, saying if the game were real life they would fire him.



and Planning

BY LIZ EINSTEIN

Should students seek the services of college faculty members as consultants, no transaction may be made until the extent of the professor's input is defined, a fee arranged and a contract made. When these faculty consultants are "hired", students then must subtract the fee from their computer budget. While no real money changes hands, students learn such information is not free in the professional world.

Appel steered his Board onto a project and dashed to a City Council meeting. Prepared and knowledgeable, students speak with expertise. Based on weekend research, Appel voiced opposition to a proposed highway his group felt would split the city and permit ghettos to develop. Since the highway was in the county's district, its construction concerned the group because of possible major long-term social and economic problems they felt it would bring.

As Appel listened to the highway proposal, a regional planner came by and whispered, "New developments have come up. Can we talk before the County Council meeting?" Appel nodded.

After the meeting, he dashed back to his "office" to check his Board's progress and then left to attend the County Council meeting. Again, outside homework paid off. A developer wanted his planned unit development okayed so construction could proceed. His presentation to the council members was elaborate and professional.

"I hope this goes through," Appel whispered. "That guy is my roommate, and you wouldn't believe the work he put into this presentation."

It was worth it. The council members discussed its merits and shortcomings and voted on the issue. It passed. And the look that passed between the developer and the planner, really students and roommates, was as genuine as if a $50 million deal had just been consummated.

Peer pressure is great. If students fail to do their homework, they face embarrassment when they lack the answers to questions posed at these meetings. But if they have red faces in this gaming situation, it's likely that as professionals they will not repeat similar mistakes.

When homework is sketchy, when incorrect information is given or when proceedings are improper, Professor Emeritus C. Eugene Farnsworth, serving as a judge, stops the session to give a mini-lesson. For instance, at a council meeting someone attempted to vote on an impact statement. Farnsworth stopped the meeting to explain that impact statements are simply informational and city or county councils cannot approve or disapprove them. Even the seating of county officials must be correct or "the judge" comes down on them.

"It's best that students make mistakes here in the classroom," said Farnsworth, "rather than blundering in the real world."

The game is taken seriously by nearly every student. Developing a professional attitude is a priority of the school and students know a strong interface exists with land planners and landscape architects. Many will be employed by planning agencies, but even those who will work in private firms will eventually make presentations to planning agencies.

To learn to operate the game and adapt it for use in the landscape curriculum, Douglas Johnston, a third-year student, traveled to California with Professor Lewis this past summer. He now serves as a role advisor to his peers, showing them ways to make the game more exciting, guiding them to the proper sources of information and helping find alternative decisions. He also manages the data collection and stays until after the gaming session ends, coding the data generated that evening for computer input. Johnston is careful not to tell his classmates what to do.

"Every role has the potential for dominating," he said. "It depends upon the student filling that role. This is exciting learning because when you can experiment like this at no cost to anyone or anything, you can take your ideas as far as you want."

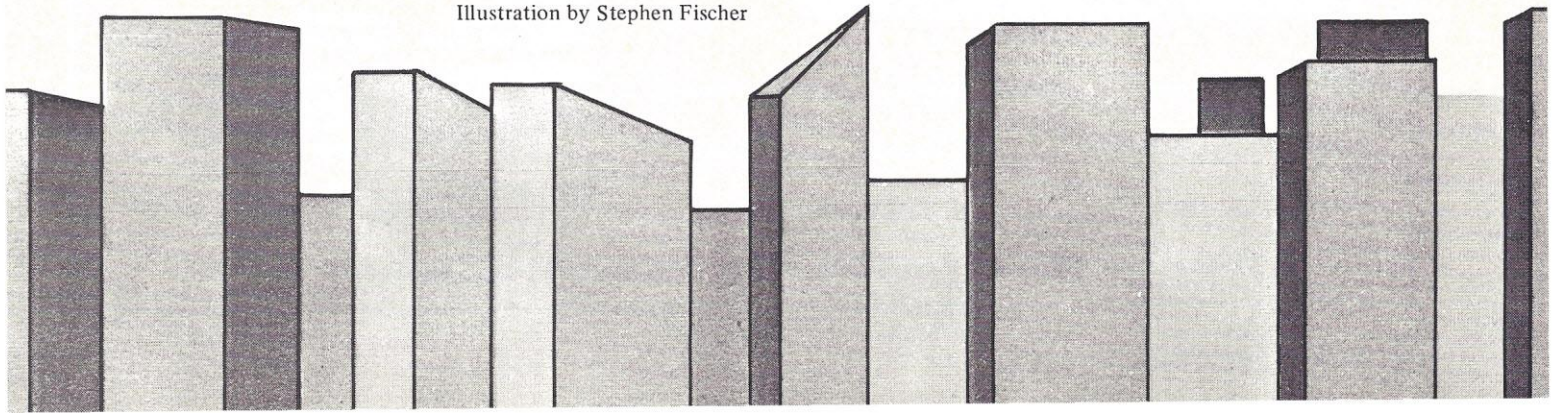
In addition to the 3-4 hour Tuesday night gaming session, three weekly lectures, papers, and exams are also required parts of the course. Although Lewis counts the game as one-half the course, many students feel the game makes it a course-and-a-half.

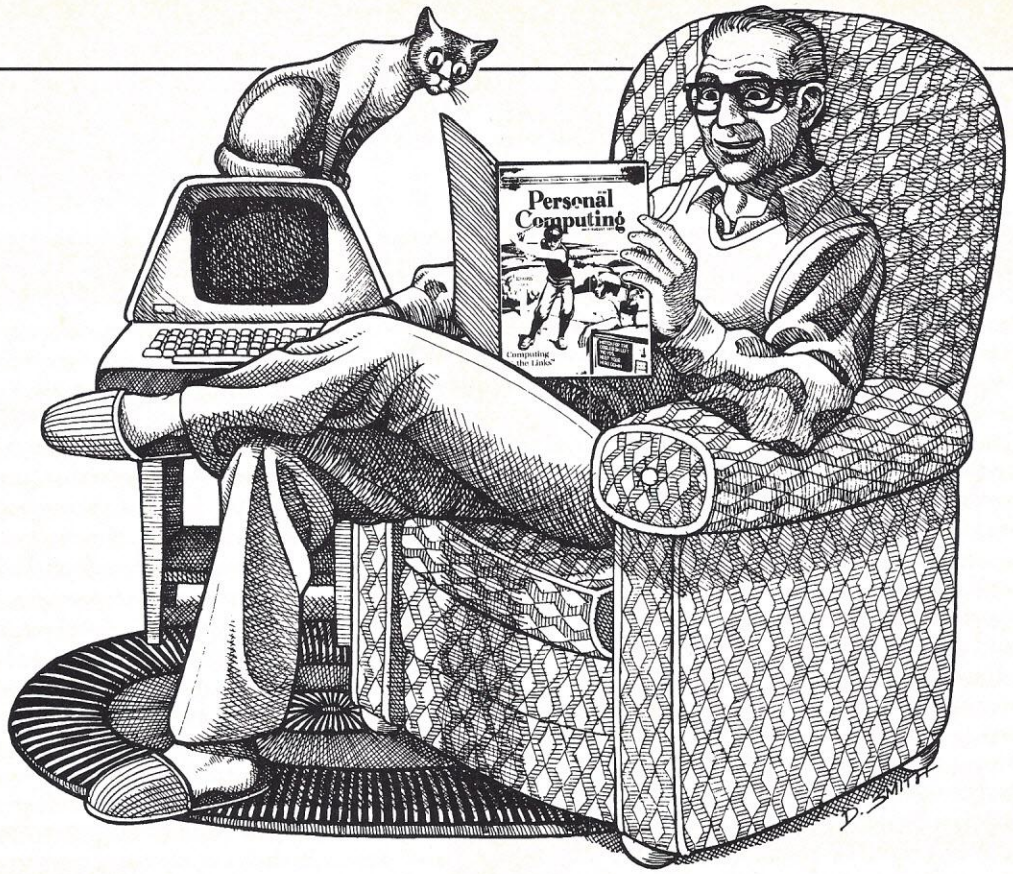
The next time around Lewis says he hopes to improve the adaptations and add more role advisors. But for a first-time effort he says it has been highly successful. Some students are thinking seriously about issues that may have turned them off in a lecture setting.

Learning to maneuver in the professional world is time-consuming and frustrating for new graduates. But with METRO-APEX to help bridge the gap between theory and the ability to apply it, ESF's landscape architect graduates will have an advantage.

Reprinted with permission from Rod Cochran, assistant to the president for community relations, State University of New York, Syracuse. Originally published in ESF '78, *January.*

Illustration by Stephen Fischer





Leading the way to computer mastery . . .

Maybe you own a computer. Maybe you're thinking about buying one. Or perhaps you're just curious about the computer world. Let *Personal Computing* show you how to use microcomputers for fun and profit.

A year's subscription brings you 12 monthly issues filled with informative, easy-to-read articles on all aspects of personal computing. Games to play, money-making ideas and applications and programs for home, business and educational use are just a few of the topics covered in *Personal Computing.*

We provide instructive articles on computer basics as well as more advanced articles.

So learn to master your computer the easy way; curl up in your favorite chair with *Personal Computing.* Use the attached coupon to order your subscription today.

Personal Computing Magazine

I've been bitten by the bug. Enter my subscription to Personal Computing today!

NAME_____

ADDRESS_____

CITY_____ STATE_____ ZIP_____

USA

☐ 1 year (12 issues) $14
☐ 2 years (24 issues) $26
☐ 3 years (36 issues) $38

☐ Bill me
☐ Check enclosed
(you'll receive one extra issue for each year!)

Charge my:
☐ Master Charge
☐ Bank Americard

Account # _____
Card expiration date _____

ADDITIONAL POSTAGE (per year)

CANADA & MEXICO $4.00 surface $8.00 air
OTHER FOREIGN $8.00 surface $36.00 air
(Please remit in US funds – Thank you)

BACK ISSUES $3.00/COPY
(Payment must accompany order.)

Mail to: PERSONAL COMPUTING, 1050 Commonwealth Ave., Boston, MA 02215

Using Tables for Machine Language Programming

BY STEVE A. HUGHES, JOE CELKO, AND ELIZABETH M. HUGHES

Those of us born in the Dark Ages (B.C. – before calculators) remember using tables many modern students have never needed, such as trigonometric tables or (before "new math") multiplication tables. Today, about the only table routinely seen in use is a sales tax table taped to a cash register. But, in computing, tables are still a valuable tool, forming one of the most convenient ways of handling and referencing related data.

The most common sort of table used in computing closely resembles the sales tax table mentioned above. It is a list of related data which begins and ends at known locations in memory and each entry has two parts: an *index* and a *datum.* The index helps the program find the datum it needs. Just as the cashier, seeking the sales tax on a purchase totalling $19.46, searches the index column for that amount and reads the appropriate tax from the column next to it, so the program searches the indices in a stored table to find the datum it requires.

As with most other programming techniques, using tables involves a time/storage trade-off; but tables pose a more delicate question than many other programming tools. If the table is short and search time is minimized, using a table can save both time and storage; but if the table is long and the search is clumsy, it can require more time and more storage than a comparable routine to compute the data needed directly. In each application, you should carefully weigh the advantages and disadvantages of using tables.

It is almost always wise to use tables where the data to be retrieved cannot be readily calculated – for example, on address lists or when converting from one computer code to another. If you expect repetitive calls on the same data, it's usually more efficient to use a table than to recalculate the desired datum each time you want it. Even when the calculations routine is easily written, it's sometimes better to replace it with a table.

Consider, for example, the problem a multiplication or division routine can pose for some microcomputers. Although it's possible to write a multiplication routine, it may be faster and easier for the program to refer to a multiplication table, especially if the task involves only a few repeated products or a constant multiplier. Any time you can replace an involved routine by a table with a simple access routine, you should consider using tables – if you have the storage needed.

Tables can also provide convenient solutions to two special problems. Creating a true random number generator is a complex task, but you can simulate random number generation well enough for most uses by entering into a table numbers in the range desired, arranged in a random order and using them in turn as you need random numbers. Still greater randomness can be attained by programming the computer to step through the table one entry at a time at each processing cycle, stopping and outputting the number at which it stopped only when there is keyboard input. This technique is used in the programs for casting I Ching, an ancient Chinese method of divination (see related story, p. 38).

Tables can also solve, readily, problems that arise when control of a process requires many operations. If, for example, you're building a system to simulate occupancy in an empty house (while on vacation, for instance), various hours of the day or night will require turning on or off numerous lights and appliances, and possibly even taperecorded sound effects (footsteps, doors closing). A repeated series of operations keyed to a realtime clock could be used, with each operation handled separately; but using tables provides a more flexible, easier approach.

Dividing the day into categories – such as "getting-up time", "mealtime", "housework time", "recreation time", "bedtime", "background(day)", and "background(night)" – allows you to use a single pointer in a table, accessed by the realtime clock, to call up the entire sequence of operations any one of these states requires, falling back on "background" operations in between specific non-background states.

Building a table is usually simple. You'll find that most compiler languages include specific table-building commands. Yet even in machine or assembly language the process isn't difficult. Where you want to allocate and empty the memory locations a table will require before the data to fill the table is available, you can (often by a single command) specify the table's starting location and fill it and the required number of subsequent locations with zeroes. If the data for the table is at hand when you set up the table, you can enter the data directly into the table's locations.

If there is a straightforward mathematical relation to generate the data, you can establish a loop to fill the table as desired.

Table 1 shows a short multiplication table set up using a loop and a constant multiplier. Notice that we've used "invisible indices" in the table. Naturally, you can use this shortcut only when the data involved are of uniform length and the indices are sequential. But when usable, invisible indices greatly reduce a table's storage requirements while easing data retrieval.

With data of uniform length, you can simply store index then data, index then data, until the table is complete. A less direct approach (but one frequently used when some of the data are identical or when they are irregular in length or some other important feature) is not to store the data in the table at all. Instead, the address where the datum can be found is stored after the datum's index. Thus, the table is filled with index then address, index then address, and so forth. This method is similar to a book's table of contents. After the chapter number, the reader sees the page on which the chapter is found, rather than the entire chapter.

Arranging a table's data properly can simplify later access. Consequently, unless the table's purpose is randomness (as in the suggested random number table), you should not leave the table disordered. For meaningful data, the relation between the items can suggest a useful order. Whether you impose this order on the data before or after they're entered in the table is of little importance; but ordering the data is almost invariably essential.

Order is important because once you set up a table, you need a way to access its indices and, thereby, its contents. Since you know the table's starting address in memory and since each index has a known location in relation to that base address, you can get a desired datum from the table by "doing the displacement" – by jumping the required number of words to reach the index and obtaining the datum from the following location.

In many uses of tables, however, the address of some particular index is not known. For example, suppose a club stores its membership list in a table with entries arranged in alphabetical order by name followed by telephone number. With a shifting membership (people joining or dropping out), you simply can't remember where any one name falls in the roster. Such a table is worthless if the only way to access its data is by referring to a known index-address. You must also be able to search the table for some particular index and, having found it, return its datum.

There are many ways of performing a search and the one chosen for any specific application will reflect the way the table is set up. Hence, you should set up the table with the search in mind.

The simplest method is called a *linear search.* Starting at the first index in the table, the program checks each index in turn, only stopping when either the desired entry is found or when the table ends without including the needed entry. This search is neat, simple, easy to write and horribly slow. Obviously, a table of n entries requires an average of $n/2$ tries to find the needed index. A long table examined by linear search can cause measurable delays despite the computer's processing speed. Consequently, you want to reduce search time by any available means.

Fortunately, there's a simple way to reduce the search time required by linear search. In the real world, some things are more likely than others. Suppose, for example, a spy wants to use his computer as a simple cipher machine – typing in his message in plain English and obtaining the printed output in cipher. If he sets up a table in alphabetical order with the alphabet as indices and their cipher equivalents as data, he'll get an unnecessarily long search time between his input of a character and output of its cipher equivalent. He could reduce search time tremendously by using a table based on the frequency with which letters occur in written English. Since E occurs most frequently, T next most frequently, and so through Z, the beginning of the table would contain the letters he would probably use most often, drastically reducing the need for the computer to search beyond the first few entries. This principle applies to many searches: if some particular entry or set of entries is more likely to be called on than the others, put it (or them) first in the table.

Changing the search rather than the table can also shorten search time. If, for example, the table is in ascending numerical order, it isn't necessary to examine every index. By examining only certain indices (say, every fifth one) to see if it's smaller or larger than the index being sought, you can move through the table far more rapidly. If the index being examined is smaller than that desired, take another step of five indices forward. When the index examined is larger than the one sought, begin a linear search from one step of five indices farther back, progressing one entry at a time.

The amount and type of data stored will determine what size step to use in the preliminary search (every second index, every tenth, every hundredth, or whatever). This technique can be effective for any ordering – alphabetical, ascending or descending numerical, or chronological.

You can improve this search with a *binary search.* Again, we assume the table is in order. In a binary search, the program first examines the middle entry. If it's not the desired one, it's either too late or too early in the table, which automatically reduces the number of entries to be searched by half. Next, the middle entry of the half which includes the index sought is examined; and, again, if it's not the desired entry, it at least halves the number of entries to be examined. This process continues until either the desired entry is found or it's shown that the table contains no such entry.

This kind of search can be implemented in either of two ways: by using pointer variables to hold the values of the top and bottom of the section (half, quarter, eight) of the table currently being searched; or by storing the location of the "middle" entry being examined and the size and direction of the next "step". There's little difference between these approaches, so which are one you choose depends entirely on convenience and personal preference.

You can take another approach to reducing search time when you know the order in which the table is arranged.

Suppose, for example, a computer club holds a small local convention and arranges with local dealers to provide 25 small computer-related prizes of various sorts to be given away to the first 25 people who register for the event. The value of the prizes varies, from a blank prototyping card valued at $3 to a book of computer games valued at $20. The convention chairman wants the 25th registrant to have as good a chance as the first of receiving a valuable prize; but to avoid suspicions of favoritism he doesn't want the responsibility of choosing which registrant gets which prize.

For a solution, he turns to his computer. As he receives each prize from a dealer, he enters its identifying information into the computer, forming a table of prizes numbered from 1-25. If the number of prizes is large enough (or if he's as busy as convention planners always are), he probably won't re-

member the order in which the prizes arrived. Then, as registrations come in, he enters them on the computer.

A simple program to progress through the table one item at a time can tell him which prize each registrant should receive. Of course, more complicated programs can be devised to guarantee that every registrant has a chance at each prize.

For practical purposes, no true search is necessary when both the incoming data and the table are in the same order, since the next table entry to be called will always follow (rather than possibly precede) the previous one.

Note, however, that there's no reason the pointer marking the entry currently being examined must always be reset to the first entry whenever a new search begins. Depending on the data and the table arrangement, it can be more efficient to let each new search begin where the last one left off, only resetting to the first entry when the end of the table is reached.

Sometimes, a search pattern going from the first entry to the last, then from the last to the first, and so on, is most efficient. And sometimes it's best to reset the pointer after each search to some location other than the first entry. You should find the most efficient search for the particular data set and table arrangement at hand. Careless choice of search can easily counteract all a table's advantages.

Special problems arise when you cannot presort the data being input to form a table. Suppose, for example, a merchant offers a product in an advertisement, giving the reader an address from which to order it. He hopes to create an address file of the people who place orders so later he can send them catalogs of his other wares. Since the orders are not likely to arrive in any useful order (alphabetical, by zipcode, or so forth), he must furnish the desired arrangement himself.

He could store the data in the order in which he received it and later sort the file into the preferred order. But if he hopes to receive a continuing series of orders for his product, there may be no natural "stopping point" when he can conveniently perform the sort. Also, since he has no way of predicting how many people meeting the requirements of some particular section of his file (same zipcode, for instance) will answer his advertisement, he can't predict how much space any particular part of the file will require. If he's using a storage medium such as magnetic disk, where average search time for locating an item will be greatly reduced if the data are evenly distributed, he has the additional problem of trying to provide even distribution in his categories – a very difficult task.

To cope with such problems, hashcoding was developed. Using this method, some part of the data to be filed (the first few digits of the street address, for example), is operated on by a complex mathematical function called a *hash function.* The result is the index under which you file the item.

Developing good hash functions is a demanding and complicated task, frequently used to satisfy master's thesis or doctoral dissertation requirements in the fields of mathematics and information sciences. So we'll only say hash functions must assure, first, that the data operated on are distributed as evenly as possible throughout the alloted storage area and, second, that no two pieces of data are filed in the same location.

100	Location	0 Contents
101	T(1)	5
102	T(2)	10
103	T(3)	15
104	T(4)	20
105	T(5)	25
106	T(6)	30
107	T(7)	35
108	T(8)	40
109	T(9)	45

Table 1 – This multiplication table uses invisible indices. Instead of filling T(1) with 1 and T(2) with 5 (the product of 1x5), T(3) with 2 and T(4) with 10 (product of 2x5), and so on, we have simply entered the products (5, 10, 15, . . . 45) one after another, this trick can save storage and reduce search time, since, when the location before the table's data begins is filled with zero, you need only look at that location (100, in this case) plus the multiplicand to find the desired answer. Thus, in the table shown above, to find the product of 5x7, one need only look at location 100 plus seven.

This latter problem, the hash codes for two different pieces of data resulting in storage at the same location, is called "hash clash", or, if you prefer, "collision". When this problem occurs, an exception routine must be called into action to provide locations in which to store the clashing data and, later, to retrieve the data from the file.

Frequently, exception routines to retrieve clashing data involve a linear search of the area surrounding or adjacent to the location of the clash, but many approaches to exception routines have been taken.

The main drawback to hashcoding data on a small system is it requires more storage space than almost any other technique. Not only must you allot the file enough space to permit storage of all the entries likely to need filing, but you must also leave "elbow room" for the hash function, since the hashcoded data will not be filed sequentially.

There will be many gaps in a table caused either by the hash function's operation to get even filing distribution among the data, or by the necessity of leaving gaps to allow space for adding additional data as you receive it. In fact, most of a hashcoded table is usually empty – a luxury few small system owners can afford.

A hashcoded table is usually an example of a *non-contiguous* table, that is, one which has gaps in it rather than being completely filled. A *contiguous* table, on the other hand, contains no gaps or blanks. Naturally, a non-contiguous table containing five items of data requires more storage than a contiguous table containing the same data, since it will also contain contain unfilled locations.

A non-contiguous table has advantages, especially when, as often happens, you want to insert an entry between two items already stored in the table. There's a horror story in computing circles about a contiguous table that filled 15 disk-packs. Then one day it became necessary to insert an additional datum early in the table. The moral, of course, is to leave some gaps if you anticipate any additions.

Often tables aren't a good idea. Simply calculated functions are best calculated rather than, for example, making an "addition table" of the sums of one-digit numbers plus one. Also avoid tables when the program seldom uses the data they contain. On some systems with extremely limited memory, the storage required by the tables may prevent their use in all but very special cases. The rule of thumb is to use tables only when they are necessary or provide clear-cut advantages.

For some tasks, though, tables fur-

nish the best or easiest answer. A business would benefit from an inventory table against which to check inquiries about items in stock, orders and deliveries. Computerists benefit from shift tables, which indicate, for each number to be converted from hex to decimal, the number of times the hexadecimal number must be shifted in a register to get its decimal equivalent.

Also handy are pointer tables, useful when you must process different kinds of data in a program differently. Their entries, indexed by the kinds of data, are pointers which direct the program to modules for dealing appropriately with the diverse data types.

Tables can also store alpha characters required by output sentences. A program can search the table for the components of any needed sentence, rather than limiting the output to preentered words.

A good table, then, represents a very favorable trade-off: you can save much time and greatly simplify the program by expending some storage. Well-designed tables access quickly, have a minimum of blanks and empty spaces, and do not store in each entry factors common to all. (For instance, if all the data in a table is added to x before being used by the program, there's no need to store x in the table.) Generally, it's unnecessary and wasteful to assign a variable name to each entry in a table, since the entries can be accessed efficiently without recourse to variable names.

Additionally, good tables are set up to prevent their unintentional use for any other purpose than as tables. Many a novice programmer has been appalled to discover that, due to his failure to significantly distinguish his table from the rest of his program (with characters or commands which are either harmless or cause recognizable error messages), his "search" continued beyond the bounds of the table to blithely chase through the rest of his program code. Worse still, this failure can permit the program to attempt to execute the table's contents as commands. A little care in establishing tables, however, prevents these problems – the advantages of using tables justifies taking the necessary preventive measures.

Like any other programming technique, table-building can be used effectively or disastrously, depending on the programmer.

So try some tables in your programs. You'll find them a convenient solution to a number of programming problems, and, with well-selected searches, a real time-saver. □

Stalking the Wild I Ching

I Ching is an ancient Chinese method of divination intended to tell the inquirer whether contemplated projects or events will turn out luckily or unluckily for him. Traditionally, I Ching uses fifty stalks of the plant *Ptarmica Sibirica,* gathered from the grave of Confucius (although scholars doubt the supposed connection between Confucius and the I Ching's interpretation). Modern usage permits almost any substitute for *Ptarmica Sibirica* stalks.

The stalks are manipulated 18 times (thrice for each line) to develop a figure of six parallel broken or unbroken lines, called a hexagram, on the basis of which the luck of the enterprise in question is determined. A hexagram may contain all broken lines, all unbroken lines, or one of the 62 combinations of broken and unbroken lines. (See Figure 1.) Additionally, sometimes manipulation of the stalks results in *moving lines.* These lines are treated as, for example, broken lines in the major hexagram; but they change their value to its reverse (unbroken) to

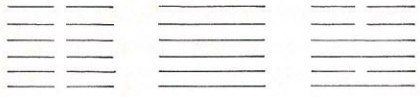


Figure 1. Hexagrams.

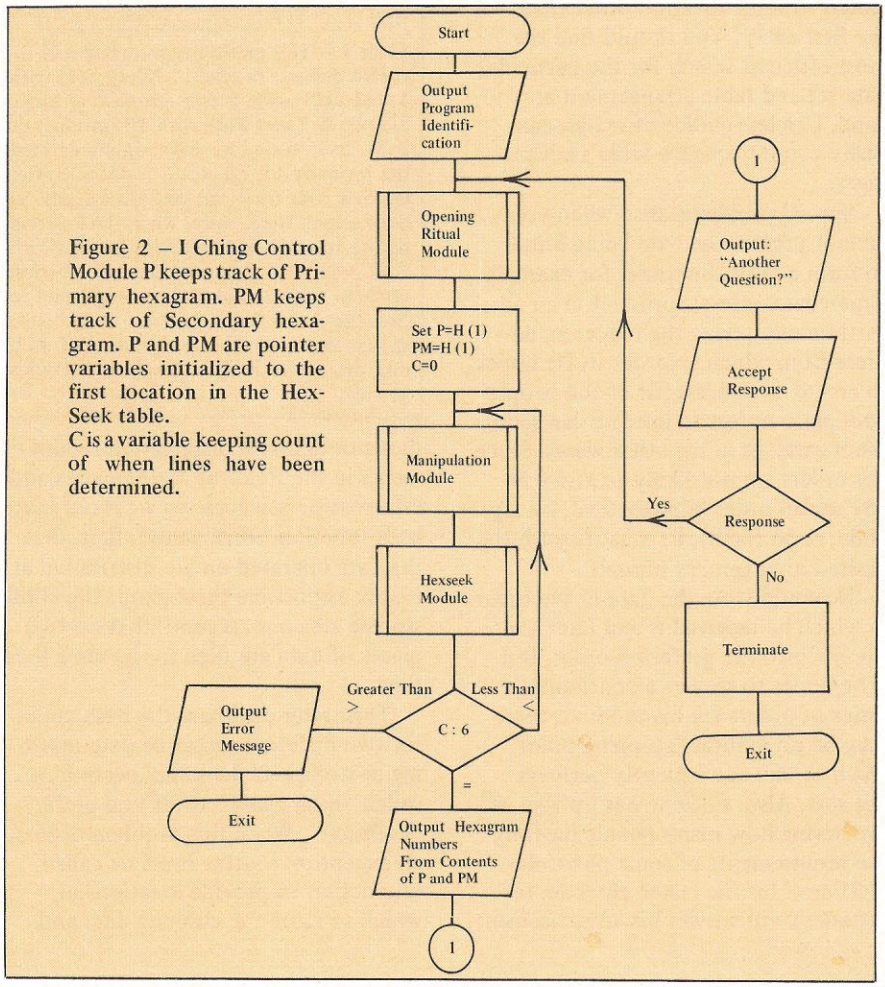


Figure 2 – I Ching Control Module P keeps track of Primary hexagram. PM keeps track of Secondary hexagram. P and PM are pointer variables initialized to the first location in the Hex-Seek table.
C is a variable keeping count of when lines have been determined.

form a secondary hexagram, interpreted as modifying the meaning of the first (major) hexagram.

During the opening ritual, the inquirer or diviner kowtows three times to the north and passes the stalks three times clockwise through the smoke of burning incense. Then one of the fifty stalks is set aside and no longer used in the process.

To manipulate the stalks, the inquirer (or diviner, if the two are not the same) separates the 49 stalks into two bundles, attempting to split the bundle of 49 stalks nearly in half. From the lefthand bundle, he removes four stalks, then four more stalks, and so on until only four or fewer stalks remain. After setting this remainder (4, 3, 2, or 1 stalks) aside, the diviner performs the same operation on the righthand bundle created by the split and adds its remainder to the remainder heap.

Then he recombines the bundles (less the ones in the remainder heap) and goes through the full process all over again, adding still more stalks to the remainder heap. Upon completion of this second manipulation, he repeats the process again, using the still smaller group of stalks not yet in the remainder heap. Upon finishing this manipulation (with its attendant additions to the remainder heap), he counts the number of stalks in the remainder heap. They will always total 13, 17, 21, or 25.

Thirteen stalks means a moving line that is unbroken in the primary hexagram and broken in the secondary hexagram; 17 means a broken line; 21 means a moving line broken in the primary hexagram and unbroken in the secondary hexagram; and 25 means an unbroken line. Now the diviner knows the shape of the lowest line in the hexagram.

This complete procedure is then repeated (beginning anew with the full 49 stalks on each full repeat) five more times to determine the second, third, fourth, fifth and highest lines in the hexagram. Once the hexagram (and, if necessary, a secondary hexagram) has been completed, the interpretation is given by reference to its shape and number in the interpretive texts of the I Ching.

We have flowcharted a program for performing the opening ritual and the stalk manipulations, since few readers will have enough storage available to load in a table of the interpretive texts to be printed out once the hexagrams have been found. Reference can more easily be made to a library copy or one of the inexpensive paperback editions of the I Ching available.

Figures 2 through 7 give annotated flowcharts for a program to cast the I Ching by computer. Happy divination!

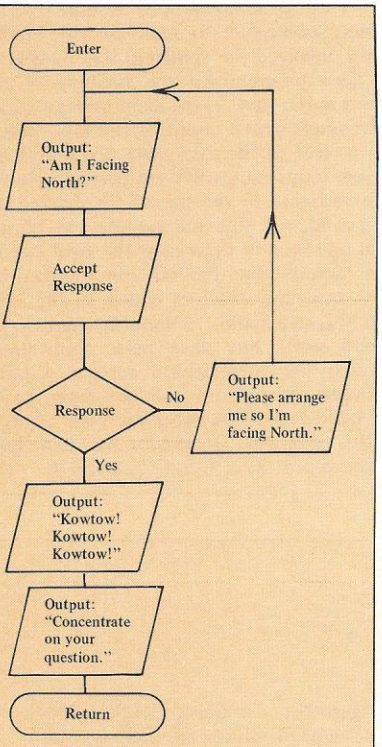


Figure 3 – I Ching Opening Ritual Module. Handling of the opening ritual is largely dependent on programmer preference, since few computers can pass the stalks through incense or determine – and voluntarily face – North unassisted. Kowtowing also poses problems for the computer. Some programmers will choose to omit this phase of the procedure; others will opt for the greater "sincerety" shown by expanding the Kowtow process and adding other routines. A basic Opening Ritual module, however, is shown here.

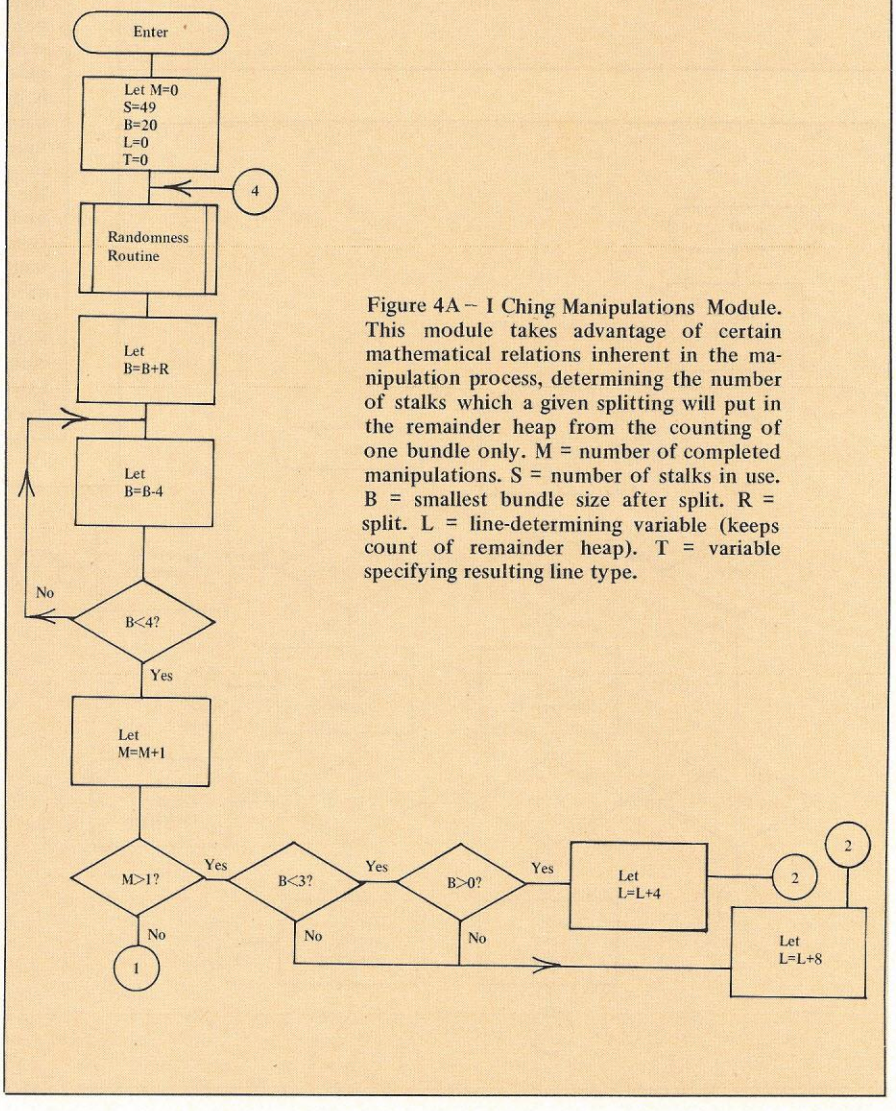
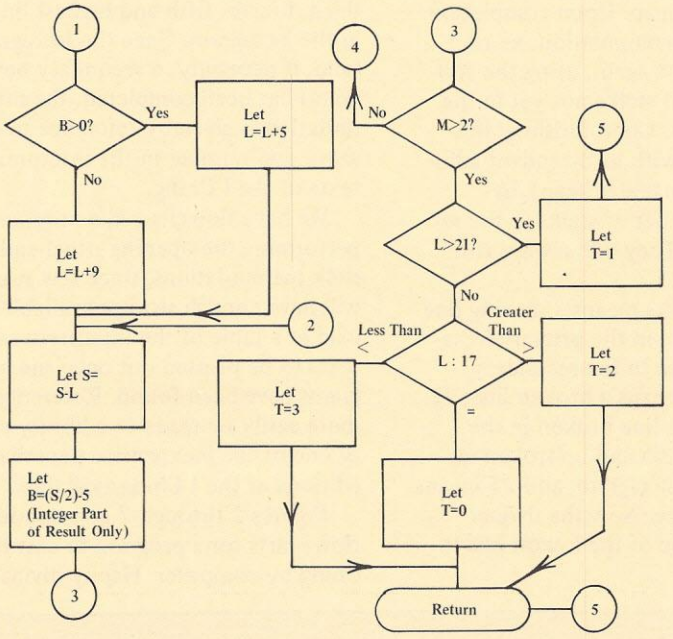


Figure 4A – I Ching Manipulations Module. This module takes advantage of certain mathematical relations inherent in the manipulation process, determining the number of stalks which a given splitting will put in the remainder heap from the counting of one bundle only. M = number of completed manipulations. S = number of stalks in use. B = smallest bundle size after split. R = split. L = line-determining variable (keeps count of remainder heap). T = variable specifying resulting line type.

Figure 4B – I Ching Manipulations Module *(cont.)*



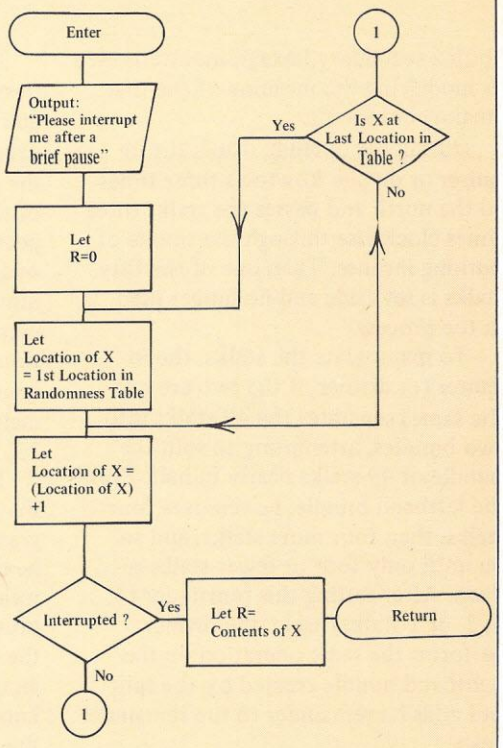


Figure 5 – I Ching Manipulations Module *(cont.)*; Randomness Submodule to replicate the human intervention involved in splitting the bundle in half. We have programmed the dividing of the stalks in two parts: first, in the main manipulations module the fundamental count of the half-splitoff is defined as half of the total stalk count (integer part of result only) minus five (to permit a plus or minus five variation in the size of the divided off bundle); second, via this sub-module, we permit the human operator to determine the final size of bundle by determining the random number between zero and nine which is added to the basic bundle. In this application, a variable changing its value would serve; but since most applications don't involve the incremented integer relation found between the random numbers here, we include the table-dependent form for your convenience in other programs. The randomness table here has ten entries: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

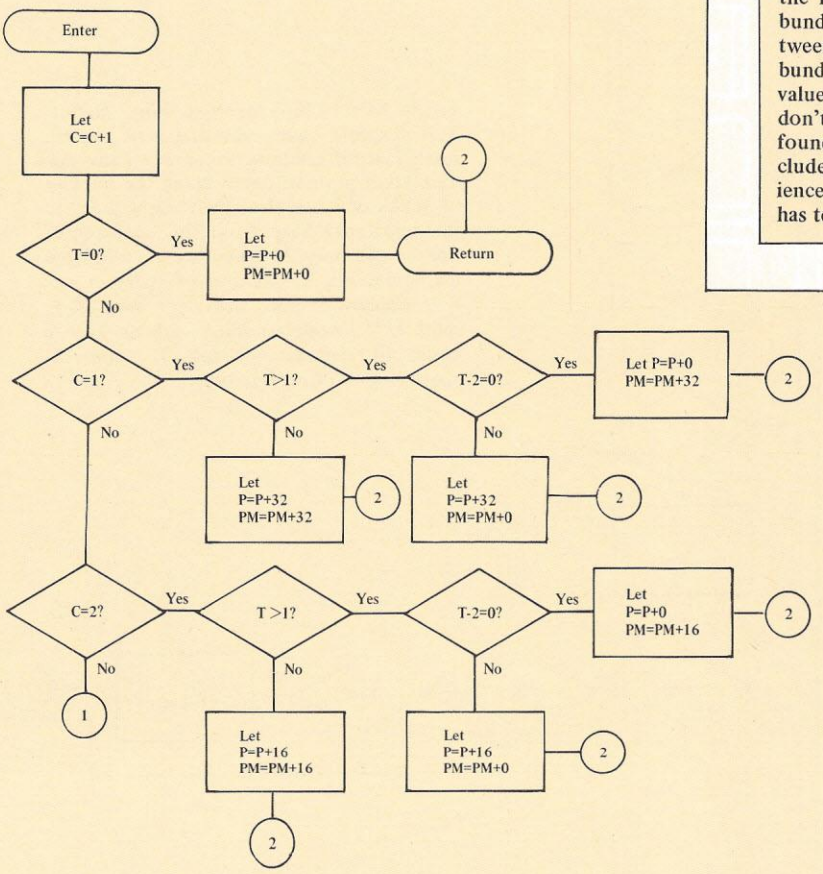


Figure 6A – I Ching HexSeek Module depends on the specially arranged HexSeek table shown in Figure 7. The variables C, P and PM are explained in Figure 2. C indicates which line of the hexagram is under consideration. P and PM give locations in the HexSeek table for ascertaining the hexagram numbers of the primary and secondary hexagrams, respectively.

Figure 6B – I Ching HexSeek Module *(cont.)*

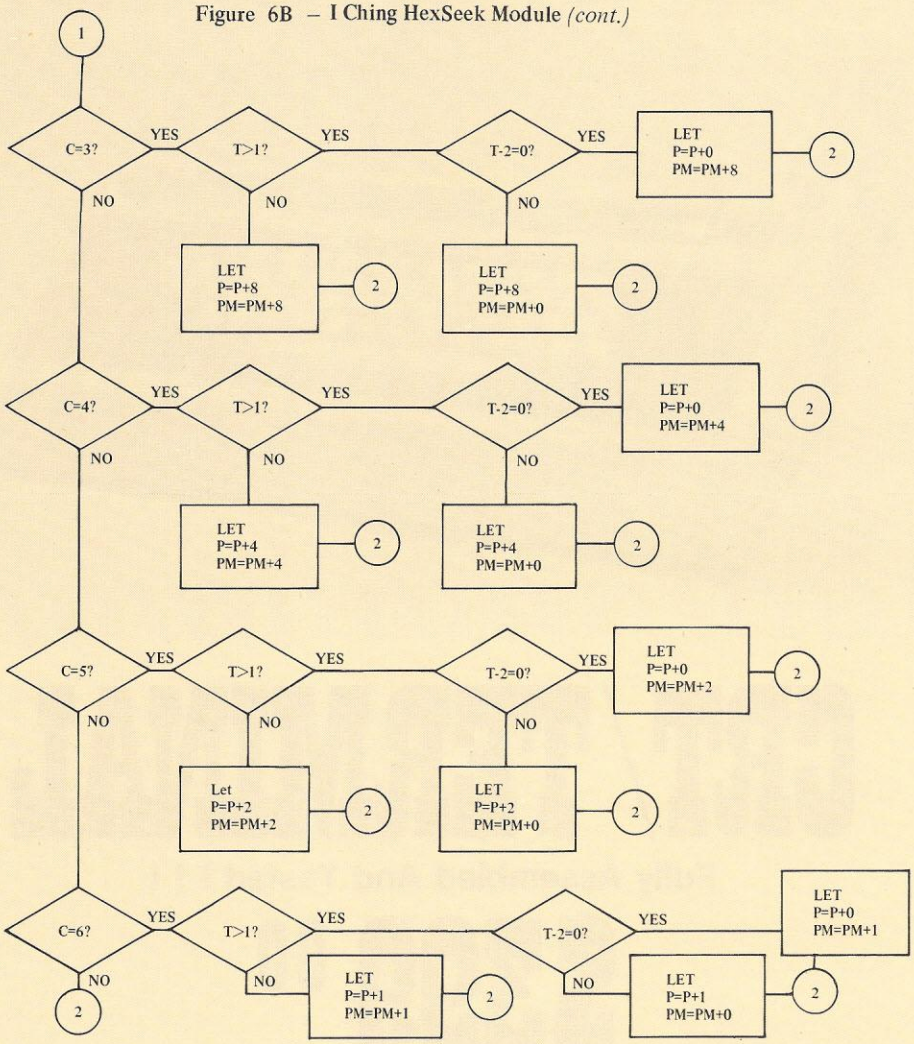


Figure 7. I Ching HexSeek Table. There are 64 possible hexagrams. This table is arranged to permit the specification of each line, as it is developed, to rule out whole groups of possible hexagrams; when the last line is given, a pointer properly moved indicates the resultant hexagram. 1s were used to represent unbroken lines and 0s were used to represent broken lines. The possible hexagrams were then divided into two groups: those whose first line (last digit in this table) was broken and those whose first line (last digit) was unbroken. These groups were similarly divided on the basis of the second line (next to last digit) and so on. When the table was fully ordered, the I Ching hexagram numbers were appended to their proper figures. To implement this table, you need only store the I Ching hexagram numbers in the order given, but we show the corresponding hexagram code (in 1s and 0s) as well so you can see how the table was ordered.

Note: As discussed above and despite appearances when only hexagram numbers are considered, *This is not a random table.* The hexseek module was designed for this table arrangement and use of any other table arrangement with it will make it inaccurate.

Code	Hexagram #
000000	2
100000	23
010000	8
110000	20
001000	16
101000	35
011000	45
111000	12
000100	15
100100	52
010100	39
110100	53
001100	62
101100	56
011100	31
111100	33
000010	7
100010	4
010010	29
110010	59
001010	40
101010	64
011010	47
111010	6
000110	46
100110	18
010110	48
110110	57
001110	32
101110	50
011110	28
111110	44

Code	Hexagram #
000001	24
100001	27
010001	3
110001	42
001001	51
101001	21
011001	17
111001	25
000101	36
100101	22
010101	63
110101	37
001101	55
101101	30
011101	49
111101	13
000011	19
100011	41
010011	60
110011	61
001011	54
101011	38
011011	58
111011	10
000111	11
100111	26
010111	5
110111	9
001111	34
101111	14
011111	43
111111	1

CRT/TERMINAL

Fully Assembled And Tested ! ! !

$399.00

STANDARD FEATURES:

16 x 32 SWITCHABLE TO 64 x 16 DISPLAY

RS232 I/O PORT

GUARANTEED FOR 90 DAYS

OPTIONS:

- **9'' OR 15'' MONITOR**
- **NUMERIC KEY PAD**
- **SPECIAL FUNCTION KEY PAD**
- **MODEM**
- **S100 BUS BACKPLANE**
- **PLUS MANY MORE - CIRCLE BINGO NUMBER FOR MORE DETAILS**

Caldwell Computer Co.

546 West Olney Avenue
Philadelphia, PA 19120
(215) 224-9133

Sam Caldwell

CIRCLE 32

Music in Your Memory

BY RODGER POGUE

If you've invited your boss for dinner you might want to put on some classical music for background. For that "special dinner guest", some easy-listening music could set the mood. When your friends stop by, they might prefer some down-home country, rock or jazz.

But letting a friend choose which album he wants to hear could mean reeling off a list of every folk album or rock record you own. Finding that one classical album you bought two years ago to impress your last boss in your 200-odd record collection would take a bit of time. But it doesn't have to.

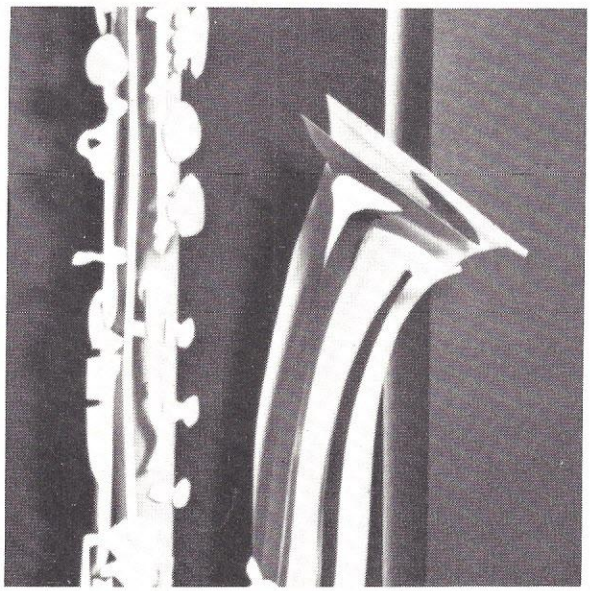
I don't know how many times *I've* gone to play a record album I'm sure I bought but can't find anywhere, or found an album that I could swear materialized out of thin air. When I look for a specific album for whatever reason, I'm lucky if I even know what room it's in. To keep track of my collection, I developed a program to sort a record album file. (Note: in this article, I use "record" to indicate a computer disk record, and "album" to indicate a music album.)

The program helps if you want to know, for example, if Lerner and Lowe wrote *Oklahoma* or if it was Rodgers and Hammerstein. Did James Taylor just sing "You've Got a Friend", or did he write it also? Which Beatle sang "Norwegian Wood"? Do you own the soundtrack from *My Fair Lady?* – wouldn't it be loverly if you could answer these questions in just a matter of minutes by locating the album immediately rather than searching through your entire album collection?

Actually, the program has many more uses. In the structure of the file, I created five sort keys in combination with data fields which may also be used as sort keys. With this program, you could easily change the fields in size or quantity, re-label the printout headings and come up with a program to sort your stamps, books, coins or photographic slides.

I keep thousands of slides in trays, but to identify a single slide could take hours. With this program you can categorize slides by subject, time, place, film or by any other method which suits your needs. You could sort your library just as easily, grouping the books by author, title, subject – again, whatever suits you best.

Or, what about your checking account? You could use the serial number on the checks and sort by medical expenses, donations, food, entertainment and so on.



Wouldn't it be loverly if you could locate any album in your collection in minutes?

Modifications to help you implement some of these suggestions appear at the end of this article.

My system is an Imsai with 48K of memory, a Soroc CRT, a DECwriter, and a Micropolis disk drive using Micropolis Extended Disk BASIC, Version 2.0. Using this system, I'm able to sort a little less than 200 albums. BASIC takes up about 22K of memory, the program takes about 14K when operating and the albums use about 12K. With the memory space and number of albums I have, the program works just right. But even if your system specs don't match mine, you can alter the program to fit your needs.

In the first section (lines 0 to 2000), the program defines its task and loads into memory the albums from the disk. In each of the next five sections (3000, 4000, 5000, 6000 and 7000) the program sorts the albums according to the category selected and then

FIGURE I

STRING

A$	ARRAY FOR ALBUM STRINGS
B$	ALBUM STRING IN UTILITY SECTION
C$	ALBUM STRING IN UTILITY SECTION
D$	CLASSIFICATION STRING
F$	FORMAT STATEMENT FOR LINE 1540
H$	DATA FIELD DELIMITER
J$	ALBUM FILE NO.
K$	ARRAY FOR ALBUM CLASSIFICATIONS
L$	TITLE KEY
M$	ARTIST KEY
N$	LABEL
O$	TITLE
P$	ARTIST
Q$	ON TAPE
R$	OTHER DATA
S$	", " STRING DELIMITER
T$	RESPONSE TO QUESTIONS ASKED IN PROGRAM
U$	COMBINED INPUT FOR ALBUM STRING
V$	NAME OF FILE TO BE CREATED
X$	SORT SWAP STRING
Y$	ARRAY FOR SORT KEYS
Z$	DISK DATA FILE NAME

NUMBERS

A	NO. OF DISK RECORDS IN FILE
A1	DISK RECORDS TO BE USED
A2	NO. OF DISK RECORDS (UTILITY)
B	LENGTH OF A$(J)
D	ARRAY FOR SORTING ORDER
E	FLAG FOR UTILITY SORT
F	VALUE OF K$
I	LOOP COUNTER
J	A$ SUBSCRIPT
K	LOOP MASTER COUNTER FOR SORTS
L	NO. OF LISTINGS
N	THE NO. OF ALBUMS IN THE FILE
N1	LENGTH OF LABEL
O1	LENGTH OF TITLE
P	PAGE COUNTER
P1	LENGTH OF ARTIST
S	FLAG FOR SORT ROUTINE
U	NO. OF PAGES
V	NO. OF ALBUMS
W	NO. OF PRINT LINES
Z	FLAG FOR UTILITY LOAD

PROGRAM NOTES

1010-1040	Dimension arrays
1050-1190	Loads array K$ with album data, clears terminal screen and displays the list for selection of different types of sorts.
1200-1290	Enters the amount and order of listings desired. Line 1280 is used to send control to either the utility or to end of job (line 1270) without using the selection process and without using array D.
1300-1360	Gets the file to be used in each run of the program. Line 1320 shows the total amount of disk records that the file has on the terminal. I included line 1330 to allow the operator to use a smaller amount of disk records than the file holds. You may wish to test certain sorts or other program functions without running through the entire file. Also, if your memory is not adequate to run the entire file, you can use this option to process the amount of albums your system can handle. Line 1335 checks for valid record number. Line 1340 interprets the amount of disk records into albums.
1400-1495	Loads the file into memory. Dimensions the A$ and Y$ arrays. Lines 1450-1455 checks for any files that have been deleted and fills in those holes. Line 1495, if this load was for the utility section, transfers control to Line 8611.
1500-1600	Clears CRT and computes the albums to be sorted. If there were any blanks in the file, this section will print the correct amount of albums in the file, the number of print lines and the amount of pages.
1600-1690	Master loop which controls the sort and printout order of the listings.
1700-1900	Clears CRT and gives an end of job.

Lines 3000 to 7000

In each of the thousand series (sections 3000, 4000, 5000, 6000 and 7000) each line in all five sections is the same except for the sort specifications and and the arrangement of the information when it is printed out; that is, in lines X120, X130, X600, X810 and X820, X represents 3, 4, 5, 6 or 7.

X000-X220	This is the sort for the particular listing.

Continued on page 46

prints the album listings. Program control returns to the master loop to perform other sorts if specified at the start of execution; if not it goes to end of job.

If you select the utility mode, the program branches to the 8000 series statements immediately and does maintenance work on the record files. Figure I lists all the variables and their definitions.

Running the Program

First, you'll get the selection list for the different types of sorts and printouts plus a utility section and an exit. If you want to sort your albums by artist, select #2. If you also want file number and title, you'd select #4 and #1 respectively, so you'd get a total of three sorts. When the program asks for "TOTAL AMOUNT OF LISTINGS" you enter "3". Next, input the three sorts you want in the order you want them printed.

The program will then ask for the file you want to sort (if you have more than one record file on the disk). After you enter the file name, the program determines if the drive is up. If not, it asks "ARE YOU READY". You need only hit return if the drive is ready. Then the number of disk records will be displayed and you'll be asked how many records you want for this running of the program.

The number displayed will show all available records containing data plus the first record, which holds the total number of disk records in the file. For example, 10 disk records, in reality, contain 9 data records plus the first record.

In the Sample Run I chose to use 3 records. (If you enter less than 2 records or more records than the file contains, you'll get an error message; just enter the correct number and hit return.)

Now, the program will load in 2 records. A message comes up stating how many albums the program is sorting, the number of print lines and the number of pages expected.

After finishing the first sort, the program will print out the albums for that sort. Then the program goes to the next sort and printout and continues until it completes all sorts. You can run this program with 1 to 5 sorts, and anywhere from 2 albums to the maximum number of albums your memory can hold.

The utility section permits file maintenance. Run the program for one listing and at "LISTING NO. 1" enter "6". The program will then branch to the utility section (lines 8000 to 9000). The utility selection list contains several choices. The first option is to either create a new file from scratch, add to an existing file or fill any blank spots in an existing file.

After you enter the file name, the program will tell you how many records are in that file. Then the program will check each disk record for either an empty record or an end of file. If you have a large file, this checking may take a few minutes. You will then be told which record you'll be adding data to.

To enter data, just follow the prompts. The data will be output to the terminal and you will be asked if it is correct. (If it's not, hit "N" and type the data again.) To end this section, enter "0". If this is the first part of a disk record, the program will ask you again to enter "0". This procedure will update the first record of the file to the current number of disk records and close the file. After you've set up a file, you'll be able to run the program with as few as 2 albums.

You can check the disk record file record by record by choosing option 2. The program will tell you the total number of disk records in the file. You input the record number you want to look at (remembering not to use record #1, which contains only the number of disk records in the file). The program then takes the record and outputs the information to the terminal.

At this point, you can either change the albums or move onto another record. There are two ways to change an album. The first and easiest is to hit "0" when asked "NEW ALBUM (A or B)". The program inserts a "0" in that file and later, after you have finished with this section, you may run the input option and enter data back into this record when the program searches for an empty file.

The other way to modify data is to type data into the record when asked "NEW ALBUM (A or B)". But be sure you enter data in the correct format.

You can print the disk file to locate bad data or to output hard copy if you choose option 3. If you use a CRT terminal for ourput, the program contains some lines which will slow up the output to the terminal so you will have a chance to read the lines on the screen.

Option 4 lets you load data files into memory, then rewrite the data to another file on a disk. This feature allows you to sort the data by a different sort key and save it on disk, or to write the data onto another disk. Also, this section automatically removes any blank albums and/or disk records.

To initialize new disk files on your disk choose option 5. This option also puts a "1" in the first record.

Options 6 and 7 have been left blank. Therefore, if you find a need for some functions useful to your particular application, you can add those functions using these option numbers.

Option 8 merely starts the program from the beginning. The last option prints out "END OF JOB" on the terminal.

Why reel off a list of every rock album you own? Let your computer do it!

PROGRAM NOTES CONTINUED

Lines X120 and X130 decode the file information and sort the data that pertains to that section. In those two lines, we are looking at a maximum of four characters per album and then determining whether or not to move the entire string, A$(J), to a new position.

X230-X320 X230 is a flag to determine whether the preceding sort was a utility sort, and if so to return to the utility section, line 8640.
Line X310 OPEN 9 "*P" ENDPAGE X800, in Micropolis, means to open a print file. The 9 represents the file the output will go to – on my system. The "endpage" is really a GOSUB statement which is activated by an internal counter, triggered by the amount of lines that are output to the printer. Each time it counts enough lines, it tabs to a new page, prints a header and returns to printing the data.
Line X320 starts the printing. I use a GOSUB statement to print out a new header at the start of output.

X400-X495 These lines decode the entire A$(J) into individual strings from X410 to X460.
Lines X465 and X475 determine the length of the label and title respectively.
Lines X470 and X480 get the actual string variables for the label and title.
Lines X485 and X490 figure the remaining string (artist) by adding the label and title strings, plus the fixed size fields and subtracting them from the length of A$(J) (the total string).
Line X495 creates a numeric value from the string value K$ for the classification category.

X580 and X590 Prints an asterisk if album is on tape.

X600 Prints out the album in a specific format for that particular sort.

X690-X700 Ends the print loop and returns to the master loop. "End page" here means to tab up to the start of the next page.

X800-X990 Subroutine to print out page headings for that section.

UTILITY LINES

8000-8300 Section menu and GOTO statement.

8300-8400 Input new data.
- 8325 Checks for end of file.
- 8335-45 Checks for blank albums.
- 8350-55 GOSUB routine for inputing data to fill blanks.
- 8362 GOSUB to input section for new files or past the previous end of file.
- 8375 Puts new data on disk.
- 8380-96 Input data routine.
- 8398 Puts new amount of disk records in record #1.

8400-8500 Modify/Delete.
- 8420-22 Check for valid record number.
- 8430-31 Prints old record.
- 8450-55 Input new data.

8500-8600 Print record file.
- 8505-14 Output to terminal or printer.
- 8549-50 A wait loop for terminal output. Line 8550 may be changed for longer or shorter waiting periods.
- 8595 Check if all data on terminal has been read before continuing.

8600-8700 Transfers data to new file.
- 8610 Goes to enter data.
- 8643-44 Tests for odd number when total albums are divided by 2. If number is odd, add .5 to it, then enter the number in the first record.
- 8660 Puts data in new file.

8700-8800 Initialize new file.
- 8720 Set up proper syntax for opening new files in Micropolis BASIC.
- 8740 Enters "1" in the first record.

8940 Clears variables and returns to main program.

8990 Go to end of job.

The File

In Micropolis, each record can be 250 characters long. In this program I can use 87 characters per album maximum, but the normal length is around 50 characters. There are two albums per disk record.

The file also contains commas at the end of both album strings to indicate end of data for the albums.

Let's go over the file record. The first four characters are the file number or serial number. In my case, I started out with 1001. Then comes the slash "/", used to delinate the data fields. The next field contains the record classification. This field could be expanded to two columns to include up to 100 possible classifications.

The next column is used for albums on tape. (I have a tape deck, and sometimes record an album onto tape.) If the album is on tape, there's a "1" in this field; if it isn't on tape, then "0" fills the field.

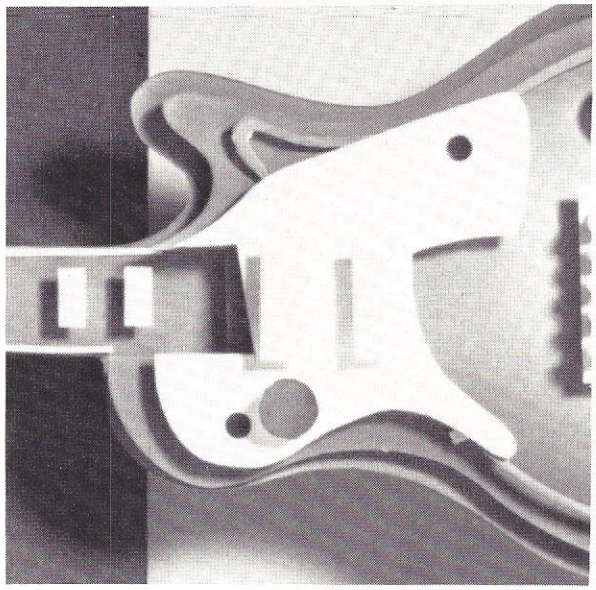
The next 3 columns have been left unassigned for you to fill in with any information you find necessary. (Of course, in the printout you will have to modify some program lines if you wish it to print.)

The next field is for the title key which describes the title, and may be part of the title. For example, "AMER" stands for album "American Graffitti". Using a sort key instead of the regular title allows for album titles that use words such as "The", "An", "A" and so on, which don't help you when trying to alphabetize the list.

Next is the artist key. Again, use the four most meaningful characters to describe the alphabetic location in the file of this album. For example, if you have an album by Janis Ian (the last name has only three letters), you'll need a fourth letter to complete the field. I use the first letter of the title key since I have more than one of her albums. Using this method allows the program to sort, within her last name, different albums in title order. However, if I had only one of her albums, and didn't expect to buy any more, the use of "J" (from her first name) might be appropriate. I also use this method for soundtrack and cast albums by labeling the artist on these "soundtrack or cast" ("SOUX" or CASX"; the 'X' stands for the first letter of the title).

The next field is for the label name. This field can run up to 10 characters long. The "LABEL SORT" section also uses the first 4 characters of this field to sort the albums by label name.

Next comes the title field, which can run up to 35 characters long and is used only for printing out the full title name in the printout. After the title comes the artist field, which is to 25 characters long and is used only in the printout. At the end of each album string is the string delimiter, a comma.



I don't know how many times I've gone to play a record I know I bought but can't find anywhere.

On Size and Space

In this program, I use all of my 48K (22K is the memory needed by Micropolis BASIC). 10K is used for the program storage. But if you are resourceful I'm sure you could use 32K or less memory.

Most BASIC versions generally use 12K or less (at least in hobby machines). You need space to put the program, but you certainly don't need more than 3K of program space for a smaller version of this program, which can give you the same output, but take more time and effort on your part manipulating the program.

To reduce program size, first decide which parts of the program you need. (Remember you're going to need the first section (lines 1000-2000) to load the files into memory.) Then, decide on which parts of the program you'll want for the sort.

With the DECwriter, I can use up to 132 print columns to print the output. But since that much room is certainly not necessary, I think you could use even a 40 column printer instead. For such a printer, your printout might consist of just the file number and any one of the other categories (e.g., the title or the artist).

So now, on the average system, you're using about 14 or 15K. Each 1K of memory can hold about 12 albums, as I have defined them. However, if you store less information about the albums, you can change the dimension statements in lines 1350 and 1400 to the largest string you'll need. No doubt you should be able to save space. The program uses about 4K of memory in active mode above the passive storage space and not including the album array.

This program can also run on a tape system. Just rewrite the utility section so you can input, update and change data sequentially rather than randomly. However, the main body of the program can be run on tape with only minor program changes. Of course, you must change the GET and PUT and other file handling statements to agree with your particular language.

The following lines should be all you have to change to implement a tape version: 1010-1040, 1310-1330 and 1440-1490. Lines 1010-1040 will need to be modified if your system does not dimension arrays as I've shown them.

To sort items other than record albums, these suggestions might help. First, decide what information you'll need in the final printout. As you decide, start to make data fields that will fit the information. Then decide in which order to arrange the fields. Now you should have a record structure that can be used by the program.

Check the different field sizes and change the dimension statements to fit them. In lines X120 and X130 you need to change the location of the sort key to make it fit

You can easily adapt this program to sort your stamps, books, coins or photographic slides.

your new file data. Lines X410-X590 in this program pull the individual fields of data out of the album string. Change them to conform to your record structure. Lines X600 and X820 output data to a printer in a specific format; change these lines to fit your requirements.

Of course, there's one more simple step necessary to make the program useful. You need to type the file or serial number of each album on a small gummed label; then affix the labels to the album jackets; and arrange the albums in order in your album rack.

Now, using your computer, you can quickly and easily locate any album in your collection. □

PROGRAM LISTING

```
LIST
1000 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1001 !                                            !
1002 !  RECORD ALBUM SORT PROGRAM 4.1978          !
1003 !  RODGER N. POGUE                           !
1004 !  REQUIRES 10K OF MEMORY(PASSIVE)           !
1005 !  MICROPOLIS BASIC VERSION 2.0              !
1006 !                                            !
1007 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
1010 S$=",":H$="/"
1020 DIMX$(87),B$(87),C$(87),K$(10,11),D(5)
1030 DIMM$(2,4),N$(2,10),O$(2,35),P$(2,25),U$(2,87)
1040 DIMJ$(2,4),D$(2,1),Q$(2,1),R$(2,3),L$(2,4)
1050 FORI=0TO9:READK$(I):NEXTI
1060 FORI=1TO15:PRINT:NEXTI
1100 PRINTTAB(20)"SELECTION LIST FOR RECORD ALBUM PROGRAM"
1110 PRINT:PRINT:PRINT
1120 PRINTTAB(25)"1 - TITLE SORT AND PRINT"
1130 PRINTTAB(25)"2 - ARTIST SORT AND PRINT"
1140 PRINTTAB(25)"3 - LABEL SORT AND PRINT"
1150 PRINTTAB(25)"4 - FILE NO. SORT AND PRINT"
1160 PRINTTAB(25)"5 - CLASSIFICATION SORT AND PRINT"
1170 PRINTTAB(25)"6 - UTILITY: UPDATES, DELETES, CREATES FILES"
1180 PRINTTAB(25)"7 - EXIT"
1190 FORI=1TO5:PRINT:NEXTI
1200 PRINTTAB(15);:INPUT"ENTER THE TOTAL AMOUNT OF LISTINGS
     DESIRED";L
1210 PRINT:PRINT
1220 FORI=1TOL
1230 PRINT:PRINTTAB(20)"FOR LISTING NO."$I;:INPUT"YOUR CHOICE
     IS ";D(I)
1270 IFD(1)=7THEN1700
1280 IFD(1)=6THEN8000
1290 NEXTI:PRINT:PRINT:PRINT
1300 PRINTTAB(15);:INPUT"WHICH DISK FILE WILL BE USED IN THIS
     SORT";Z$
1310 OPEN1Z$ERROR1315:GET1RECORD1A:PRINT:GOTO1320
1315 PRINT:PRINTERR$:INPUT"ARE YOU READY";T$:GOTO1300
1320 PRINTTAB(20)A;" DISK RECORDS IN THIS FILE"
1330 PRINT:PRINTTAB(20);:INPUT"NO. OF DISK RECORDS TO BE USED"$A1
1335 IFA1>1ANDA1<(A+1)THEN1339
1336 PRINT:PRINT"* * INVALID INPUT * *":PRINT
1337 PRINT"RE-ENTER # OF DISK RECORDS":PRINT"NOTE: IT MUST BE
     LESS THAN OR"
1338 PRINT"EQUAL TO THE DISK RECORDS IN FILE":PRINT"AND MORE
     THAN 1 RECORD":PRINT:GOTO1330
1339 A=A1
1340 N=(A*2)-2
1360 J=1:P=1
1400 IFJ<2THEN9000
1410 !
1420 FORI=2TOA
1440 GET1RECORD1A$(J),A$(J+1)
1450 IFA$(J)="0"THENA$(J)=A$(J+1):J=J-1
1455 IFA$(J+1)="0"THENJ=J-1
1460 J=J+2
1480 NEXTI
1485 J=J-1:N=J:V=N
1490 CLOSE1
1495 IFZ=1GOTO8611
1500 W=N*L:U=INT((((N/19*2)+N)/21)+1)*L:F$="ZZZV.9"
1510 FORI=1TO15:PRINT:NEXTI
1520 PRINTTAB(23)"* NOW SORTING"$N$"ALBUMS *":PRINT
1530 PRINT:PRINTTAB(17)"THERE WILL BE APPROX. "$W$" PRINT LINES."
1540 PRINT:PRINTTAB(20)"THERE WILL BE APPROX. "$FMT(U,F$)$" PAGES."
1550 FORI=1TO10:PRINT:NEXTI
1600 FORK=1TOL
1650 ONO(K)GOTO3000,4000,5000,6000,7000,8000,9000
1680 REM
1690 NEXTK
1700 PRINT:PRINT:PRINTTAB(30)"END OF JOB"
1720 FORI=1TO15:PRINT:NEXTI
1980 STOP
1990 END
3000 !SUBROUTINE FOR TITLE SORT AND PRINTOUT
3010 V=N
3020 V=V-1:S=0
3100 FORJ=1TOV
3120 Y$(J)=MID$(A$(J),14,4)
3130 Y$(J+1)=MID$(A$(J+1),14,4)
3140 IFY$(J)<=Y$(J+1)THEN3200
3160 LETX$=A$(J)
3170 LETA$(J)=A$(J+1)
3180 LETA$(J+1)=X$
3190 LETS=1
3200 NEXTJ
3220 IFS=1THEN3020
3230 IFE=1GOTO8640
3310 OPEN9"*P"ENDPAGE3800
3320 GOSUB3800
3400 FORJ=1TON
3410 K$=MID$(A$(J),6,1)
3420 J$=LEFT$(A$(J),4)
3430 Q$=MID$(A$(J),8,1)
3440 R$="0"
3450 L$=MID$(A$(J),14,4)
3460 M$=MID$(A$(J),19,4)
3465 N1=INDEX(MID$(A$(J),24),H$)
3470 N$=MID$(A$(J),24,N1-1)
3475 O1=INDEX(MID$(A$(J),24+N1),H$)
3480 O$=MID$(A$(J),24+N1,O1-1)
3485 P1=23+N1+O1:B=LEN(A$(J))
3490 P$=RIGHT$(A$(J),B-P1)
3495 F=VAL(K$)
3580 IFQ$="0"THENQ$=""
3590 IFQ$="1"THENQ$="*"
3600 PUT9TAB(5)O$;TAB(40)P$;TAB(70)N$;TAB(89)K$(F);TAB(108)"0";
     J$;TAB(125)Q$:PUT 9:PUT9
3680 IFJ=NTHEN3695
3690 NEXTJ
3695 ENDPAGE9
3697 CLOSE9
3700 GOTO1680
3800 PUT9:PUT9:PUT9:PUT9
3810 PUT9TAB(25)"*   *   *   *   * A L B U M   L I S T I N G
     B Y   T I T L E   *   *   *   *   *"
3815 PUT9TAB(123)"PAGE "$P:P=P+1:PUT9
3820 PUT9TAB(10)"TITLE"TAB(45)"ARTIST"TAB(71)"LABEL"TAB(86)"CLASSI
     FICATION"TAB(107)"FILE NO."TAB(122)"ON TAPE":PUT9
3990 RETURN
4000 !SUBROUTINE FOR ARTIST SORT AND PRINTOUT
4010 V=N
4020 V=V-1:S=0
4100 FORJ=1TOV
4120 Y$(J)=MID$(A$(J),19,4)
4130 Y$(J+1)=MID$(A$(J+1),19,4)
4140 IFY$(J)<=Y$(J+1)THEN4200
4160 LETX$=A$(J)
4170 LETA$(J)=A$(J+1)
4180 LETA$(J+1)=X$
4190 LETS=1
4200 NEXTJ
4220 IFS=1THEN4020
4230 IFE=2GOTO8640
4310 OPEN9"*P"ENDPAGE4800
4320 GOSUB4800
4400 FORJ=1TON
4410 K$=MID$(A$(J),6,1)
4420 J$=LEFT$(A$(J),4)
4430 Q$=MID$(A$(J),8,1)
4440 R$="0"
4450 L$=MID$(A$(J),14,4)
4460 M$=MID$(A$(J),19,4)
4465 N1=INDEX(MID$(A$(J),24),H$)
4470 N$=MID$(A$(J),24,N1-1)
4475 O1=INDEX(MID$(A$(J),24+N1),H$)
4480 O$=MID$(A$(J),24+N1,O1-1)
4485 P1=23+N1+O1:B=LEN(A$(J))
4490 P$=RIGHT$(A$(J),B-P1)
4495 F=VAL(K$)
4580 IFQ$="0"THENQ$=""
4590 IFQ$="1"THENQ$="*"
4600 PUT9TAB(5)P$;TAB(33)O$;TAB(69)N$;TAB(87)K$(F);TAB(108)"0";
     J$;TAB(125)Q$:PUT 9:PUT9
4680 IFJ=NTHEN4695
4690 NEXTJ
4695 ENDPAGE9
```

```
4697 CLOSE9
4700 GOTO1680
4800 PUT9:PUT9:PUT9:PUT9
4810 PUT9TAB(25)"*  *  *  *  * ALBUM LISTING
     BY ARTIST * * * * *"
4815 PUT9TAB(123)"PAGE ";P:P=P+1:PUT9
4820 PUT9TAB(10)"ARTIST"TAB(40)"TITLE"TAB(70)"LABEL"TAB(83)
     "CLASSIFICATION"TAB(107)"FILE NO."TAB(122)"ON TAPE":PUT9
4990 RETURN
5000 ! SUBROUTINE FOR LABEL SORT
5010 V=N
5020 V=V-1:S=0
5100 FORJ=1TOV
5120 Y$(J)=MID$(A$(J),24,4)
5130 Y$(J+1)=MID$(A$(J+1),24,4)
5140 IFY$(J)<=Y$(J+1)THEN5200
5160 LETX$=A$(J)
5170 LETA$(J)=A$(J+1)
5180 LETA$(J+1)=X$
5190 LETS=1
5200 NEXTJ
5220 IFS=1THEN5020
5230 IFE=3GOTO8640
5310 OPEN9"*P"ENDPAGE5800
5320 GOSUB5800
5400 FORJ=1TON
5410 K$=MID$(A$(J),6,1)
5420 J$=LEFT$(A$(J),4)
5430 Q$=MID$(A$(J),8,1)
5440 R$="0"
5450 L$=MID$(A$(J),14,4)
5460 M$=MID$(A$(J),19,4)
5465 N1=INDEX(MID$(A$(J),24),H$)
5470 N$=MID$(A$(J),24,N1-1)
5475 O1=INDEX(MID$(A$(J),24+N1),H$)
5480 O$=MID$(A$(J),24+N1,O1-1)
5485 P1=23+N1+O1:B=LEN(A$(J))
5490 P$=RIGHT$(A$(J),B-P1)
5495 F=VAL(K$)
5580 IFQ$="0"THENQ$=""
5590 IFQ$="1"THENQ$="*"
5600 PUT9TAB(3)N$;TAB(15)P$;TAB(45)O$;TAB(86)K$(F);TAB(108)"0";
     J$;TAB(125)Q$:PUT 9:PUT9
5680 IFJ=NTHEN5695
5690 NEXTJ
5695 ENDPAGE9
5697 CLOSE9
5700 GOTO1680
5800 PUT9:PUT9:PUT9:PUT9
5810 PUT9TAB(25)"*  *  *  *  * ALBUM LISTING
     BY LABEL * * * * *"
5815 PUT9TAB(123)"PAGE ";P:P=P+1:PUT9
5820 PUT9TAB(4)"LABEL"TAB(24)"ARTIST"TAB(56)"TITLE"TAB(83)
     "CLASSIFICATION"TAB(107)"FILE NO."TAB(122)"ON TAPE":PUT9
5990 RETURN
6000 !SUBROUTINE FOR FILE NO. SORT AND PRINTOUT
6010 V=N
6020 V=V-1:S=0
6100 FORJ=1TOV
6120 Y$(J)=LEFT$(A$(J),4)
6130 Y$(J+1)=LEFT$(A$(J+1),4)
6140 IFY$(J)<=Y$(J+1)THEN6200
6160 LETX$=A$(J)
6170 LETA$(J)=A$(J+1)
6180 LETA$(J+1)=X$
6190 LETS=1
6200 NEXTJ
6220 IFS=1THEN6020
6230 IFE=4GOTO8640
6310 OPEN9"*P"ENDPAGE6800
6320 GOSUB6800
6400 FORJ=1TON
6410 K$=MID$(A$(J),6,1)
6420 J$=LEFT$(A$(J),4)
6430 Q$=MID$(A$(J),8,1)
6440 R$="0"
6450 L$=MID$(A$(J),14,4)
6460 M$=MID$(A$(J),19,4)
6465 N1=INDEX(MID$(A$(J),24),H$)
6470 N$=MID$(A$(J),24,N1-1)
6475 O1=INDEX(MID$(A$(J),24+N1),H$)
6480 O$=MID$(A$(J),24+N1,O1-1)
6485 P1=23+N1+O1:B=LEN(A$(J))
6490 P$=RIGHT$(A$(J),B-P1)
6495 F=VAL(K$)
6580 IFQ$="0"THENQ$=""
6590 IFQ$="1"THENQ$="*"
6600 PUT9TAB(5);"0";J$;TAB(20)O$;TAB(55)P$;TAB(86)N$;TAB(106)K$
     (F);TAB(125)Q$:PUT9:PUT9
6690 NEXTJ:ENDPAGE9:CLOSE9
6700 GOTO1680
6800 PUT9:PUT9:PUT9:PUT9
6810 PUT9TAB(25)"*  *  *  *  * ALBUM LISTING
     BY FILE NO. * * * * *"
6815 PUT9TAB(123)"PAGE ";P:P=P+1:PUT9
6820 PUT9TAB(4)"FILE NO."TAB(25)"TITLE"TAB(60)"ARTIST"TAB(87)
     "LABEL"TAB(102)"CLASSIFICATION"TAB(122)"ON TAPE":PUT9
6990 RETURN
7000 !SUBROUTINE FOR STYLE CLASSIFICATION AND PRINT OUT
7010 V=N
7020 V=V-1:S=0
7100 FORJ=1TOV
7120 Y$(J)=MID$(A$(J),6,1)
7130 Y$(J+1)=MID$(A$(J+1),6,1)
7140 IFY$(J)<=Y$(J+1)THEN7200
7160 LETX$=A$(J)
7170 LETA$(J)=A$(J+1)
7180 LETA$(J+1)=X$
7190 LETS=1
7200 NEXTJ
7220 IFS=1THEN7020
7230 IFE=5GOTO8640
7310 OPEN9"*P"ENDPAGE7800
7320 GOSUB7800
7400 FORJ=1TON
7410 J$=LEFT$(A$(J),4)
7420 K$=MID$(A$(J),6,1)
7430 Q$=MID$(A$(J),8,1)
7440 R$="0"
7450 L$=MID$(A$(J),14,4)
7460 M$=MID$(A$(J),19,4)
7465 N1=INDEX(MID$(A$(J),24),H$)
7470 N$=MID$(A$(J),24,N1-1)
7475 O1=INDEX(MID$(A$(J),24+N1),H$)
7480 O$=MID$(A$(J),24+N1,O1-1)
7485 P1=23+N1+O1:B=LEN(A$(J))
7490 P$=RIGHT$(A$(J),B-P1)
7495 F=VAL(K$)
7580 IFQ$="0"THENQ$=""
7590 IFQ$="1"THENQ$="*"
7600 PUT9TAB(5)K$(F);TAB(26)O$;TAB(61)P$;TAB(93)N$;TAB(111)"0";
     J$;TAB(125)Q$:PUT 9:PUT9
7680 IFJ=NTHEN7695
7690 NEXTJ
7695 ENDPAGE9
7697 CLOSE9
7700 GOTO1680
7800 PUT9:PUT9:PUT9:PUT9
7810 PUT9TAB(25)"*  *  *  *  *  ALBUM LISTING
     BY CLASSIFICATION * * * * *"
7815 PUT9TAB(123)"PAGE ";P:P=P+1:PUT9
7820 PUT9TAB(1)"CLASSIFICATION"TAB(30)"TITLE"TAB(65)"ARTIST"TAB
     (94)"LABEL"TAB(11 0)"FILE NO."TAB(122)"ON TAPE":PUT9
7990 RETURN
8000 !     UTILITY SECTION
8100 FORI=1TO8:PRINT:NEXTI
8110 PRINTTAB(20)"1 - INPUT NEW DATA"
8111 PRINTTAB(20)"2 - EXAMINE/DELETE DISK FILES"
8112 PRINTTAB(20)"3 - PRINT DISK FILES"
8113 PRINTTAB(20)"4 - TRANSFER FILES"
8114 PRINTTAB(20)"5 - INITIALIZE NEW FILE"
8117 PRINTTAB(20)"8 - RETURN TO MAIN PGM."
8118 PRINTTAB(20)"9 - EXIT PROGRAM"
8140 PRINT:PRINT:PRINT
8145 PRINTTAB(20);:INPUT"WHICH OPTION";E
8148 FORI=1TO25:PRINT:NEXTI
8150 ONEGOTO8300,8400,8500,8600,8700,8800,8900,8940,8990
8300 !      INPUT DATA
8305 A1=1
8315 PRINTTAB(15);:INPUT"ENTER DISK FILE NAME FOR INPUT";Z$:OPEN
     1Z$ERROR8316:GOT O8320
8316 PRINT:PRINTERR$:INPUT"ARE YOU READY";T$:GOTO8315
8320 GET1RECORD1A2:PRINT:PRINT"THERE ARE  ";A2;" RECORDS IN
     THIS FILE"
8325 IFA1=A2THEN8350
8330 A1=A1+1:J=0
8335 GET1RECORDA1U$(1),U$(2)
8340 IFU$(1)="0"THENJ=1:GOTO8361
8341 IFU$(2)="0"THENJ=2:GOTO8361
8345 IFJ=0GOTO8325
8346 GOTO8325
8350 A2=A2+1:PRINT:PRINT"  RECORD NO. ";A2:PRINT
8355 FORJ=1TO2:PRINT:PRINT:GOSUB8380:NEXTJ
8360 A1=A2:GOTO8375
8361 PRINT:PRINT"  RECORD NO. ";A1
8362 GOSUB8380
8365 IFU$(J)<>"0"THEN8375
8366 IFJ=2THEN8375
8370 J=0:GOTO8341
8375 PUT1RECORDA1U$(1)+S$+U$(2)+S$
8377 IFU$(J)="0"THEN8398
8379 GOTO8341
8380 T$=""
8381 PRINT"IF THERE IS NO MORE INPUT, AT THIS POINT":PRINT"ENTER
     '0' WHEN ASKED FOR FILE NO."
8382 INPUT"FILE NO. @ XXXX.........";J$(J):IFJ$(J)="0"THENU$(J)=
     "0":GOTO8396
8383 INPUT"CLASSIFICATION @ X.......";D$(J)
8384 INPUT"ON TAPE (1=YES) @ X......";Q$(J)
8385 INPUT"OTHER @ XXX.............";R$(J)
8386 INPUT"TITLE KEY @ XXXX.........";L$(J)
8387 INPUT"ARTIST KEY @ XXXX........";M$(J)
8388 INPUT"LABEL: UP TO 10 CHARS....";N$(J)
8389 INPUT"TITLE: UP TO 35 CHARS....";O$(J)
8390 INPUT"ARTIST: UP TO 25 CHARS...";P$(J)
8391 U$(J)=J$(J)+H$+D$(J)+H$+Q$(J)+H$+R$(J)+H$+L$(J)+H$+M$(J)+H$+
     N$(J)+H$+O$(J)+ H$+P$(J)
8392 PRINT:PRINT:PRINT"FILE NO.....";J$(J):PRINT"CLASS.......";D$
     (J):PRINT"ON TAPE.....";Q$(J):PRINT"TITLE KEY...";L$(J)
8393 PRINT"ARTIST KEY..";M$(J):PRINT"LABEL.......";N$(J):PRINT
     "TITLE.......";O$( J):PRINT"ARTIST......";P$(J):PRINT:PRINT
8394 INPUT"IF THIS IS INCORRECT HIT 'N', OTHERWISE JUST HIT
     'RETURN'";T$
8395 IFT$="N"THENT$="0":GOTO8382
8396 RETURN
8398 PUT1RECORD1A2:CLOSE1
8399 GOTO8100
8400 !      EXAMINE/DELETE
8401 INPUT"ENTER DISK FILE NAME-TO BE MODIFIED OR DELETED";Z$
8405 OPEN1Z$ERROR8406:GET1RECORD1A:GOTO8410
8406 PRINT:PRINTERR$:INPUT"ARE YOU READY";T$:GOTO8401
8410 PRINT:PRINT"THERE ARE ";A;" RECORDS IN ";Z$;" FILE."
8415 PRINT:PRINT
8416 PRINT"IF ALBUM 'A' OR 'B' IS TO BE CHANGED, ENTER '0'"
8417 PRINT"OTHERWISE JUST HIT 'RETURN' WHEN ASKED 'NEW ALBUM'."
8419 PRINT:PRINT
8420 INPUT"ENTER RECORD NO.";A1:IFA1>1ANDA1<(A+1)THEN8425
8422 PRINT"NOT A VALID RECORD NO.":GOTO8420
8425 GET1RECORDA1B$,C$
8429 PRINT
8430 PRINT"  ALBUM 'A': ";B$
8431 PRINT"  ALBUM 'B': ";C$
8432 PRINT
8435 INPUT"CHANGE";T$
8440 IFT$="N"THEN8480
8445 PRINT
8450 INPUT"NEW ALBUM 'A' ";B$
8455 INPUT"NEW ALBUM 'B' ";C$
8456 PRINT
8460 PUT1RECORDA1B$+S$+C$+S$
8480 INPUT"ANYMORE";T$
8485 IFT$="Y"THEN8420
```

Continued on following page

```
8490 CLOSE1
8499 GOTO8100
8500 !       PRINT FILE
8501 INPUT"ENTER FILE NAME TO BE PRINTED";Z$:OPEN1Z$ERROR8502:GOTO8505
8502 PRINT:PRINTERR$:INPUT"ARE YOU READY";T$:GOTO8501
8505 INPUT"ENTER P OR T FOR PRINTER OR TERMINAL OUTPUT";T$
8510 IFT$="P"OPEN9"*P"
8511 IFT$="T"OPEN9"*T"
8515 GET1RECORD1A:N=(A*2)-2
8520 PUT9:PUT9"THERE ARE ";N;" ALBUMS IN THE '";Z$;"' FILE."
8525 PUT9
8530 FORI=2TOA
8535 GET1RECORDIB$,C$
8540 PUT9TAB(5)"RECORD NO.";I
8545 PUT9TAB(2)B$
8546 PUT9TAB(2)C$
8549 IFT$="P"THEN8554
8550 FORH=1TO100:NEXTH
8554 PUT9
8555 NEXTI
8560 IFT$="T"THEN8590
8580 ENDPAGE9
8590 CLOSE1:CLOSE9
8595 IFT$<>"P"INPUT"ARE YOU READY";T$
8599 GOTO8100
8600 !       TRANSFER
8601 Z=1
8605 PRINTTAB(15);:INPUT"ENTER NAME OF FILE TO BE TRANSFERED";Z$
8606 PRINT
8607 PRINTTAB(15);:INPUT"ENTER NAME OF FILE TO BE CREATED";V$
8610 GOTO1310
8611 !
8615 PRINT:PRINT:PRINT
8616 PRINTTAB(20)"1 - TITLE SORT"
8617 PRINTTAB(20)"2 - ARTIST SORT"
8618 PRINTTAB(20)"3 - LABEL SORT"
8619 PRINTTAB(20)"4 - FILE NO. SORT"
8620 PRINTTAB(20)"5 - CLASS. SORT"
8621 PRINT:PRINT
8625 PRINTTAB(18);:INPUT"ENTER # FOR SORT";E
8630 ONEGOTO3000,4000,5000,6000,7000
8640 OPEN2V$
8641 A$(N+2)="0"
8642 N=(N/2)+1
8643 IFFRAC(N)=0THEN8645
8644 N=N+.5
8645 PUT2RECORD1N:J=1
8650 FORI=2TON
8660 PUT2RECORDIA$(J)+S$+A$(J+1)+S$:J=J+2
8665 NEXTI
8690 CLOSE2
8695 GOTO8100
8700 !       INITIALIZE NEW FILE
8710 INPUT"ENTER THE NAME OF THE FILE YOU INITIALIZE";Z$
8720 G$="N:":Z$=G$+Z$
8730 OPEN1Z$ERROR8731:A=1:GOTO8740
8731 PRINT:PRINTERR$:INPUT"ARE YOU READY";T$:GOTO8710
8740 PUT1RECORD1A
8750 CLOSE1
8760 INPUT"MORE";T$
8770 IFT$="N"GOTO8100
8780 GOTO8710
8940 RUN
8990 GOTO1700
9000 DIMY$(N+2,4),A$(N+2,87):GOTO1410
9010 DATA"BLUES","CAST","CLASSICAL","FOLK","JAZZ","OLDIES",
     "POPULAR","ROCK","SOUNDTRACK","SPOKEN WORD"
```

SAMPLE RUN

```
SELECTION LIST FOR RECORD ALBUM PROGRAM

                1 - TITLE SORT AND PRINT
                2 - ARTIST SORT AND PRINT
                3 - LABEL SORT AND PRINT
                4 - FILE NO. SORT AND PRINT
                5 - CLASSIFICATION SORT AND PRINT
                6 - UTILITY: UPDATES, DELETES, CREATES FILES
                7 - EXIT

ENTER THE TOTAL AMOUNT OF LISTINGS DESIRED? 1

     FOR LISTING NO. 1 YOUR CHOICE IS ? 6

                1 - INPUT NEW DATA
                2 - EXAMINE/DELETE DISK FILES
                3 - PRINT DISK FILES
                4 - TRANSFER FILES
                5 - INITIALIZE NEW FILE
                8 - RETURN TO MAIN PGM.
                9 - EXIT PROGRAM

                WHICH OPTION? 5

ENTER THE NAME OF THE FILE YOU INITIALIZE? ALBUMS
MORE? N

                1 - INPUT NEW DATA
                2 - EXAMINE/DELETE DISK FILES
                3 - PRINT DISK FILES
                4 - TRANSFER FILES
                5 - INITIALIZE NEW FILE
                8 - RETURN TO MAIN PGM.
                9 - EXIT PROGRAM

                WHICH OPTION? 1

            ENTER DISK FILE NAME FOR INPUT? ALBUMS

THERE ARE   1  RECORDS IN THIS FILE

  RECORD NO.  2

IF THERE IS NO MORE INPUT, AT THIS POINT
ENTER '0' WHEN ASKED FOR FILE NO.
FILE NO. @ XXXX.....? 1001
CLASSIFICATION @ X.....? 8
ON TAPE (1=YES) @ X.....? 1
OTHER @ XXX.....? 000
TITLE KEY @ XXXX.....? MARY
ARTIST KEY @ XXXX.....? SOUM
LABEL: UP TO 10 CHARS....? BUENA VISTA
TITLE: UP TO 35 CHARS....? MARY POPPINS
ARTIST: UP TO 25 CHARS...? SOUNDTRACK


FILE NO.....1001
CLASS.....8
ON TAPE.....1
TITLE KEY...MARY
ARTIST KEY..SOUM
LABEL.....BUENA VIST
TITLE.....MARY POPPINS
ARTIST.....SOUNDTRACK


IF THIS IS INCORRECT HIT 'N', OTHERWISE JUST HIT 'RETURN'? N
FILE NO. @ XXXX.....? 110___1001
CLASSIFICATION @ X.....? 8
ON TAPE (1=YES) @ X.....? 1
OTHER @ XXX.....? 000
TITLE KEY @ XXXX.....? MARY
ARTIST KEY @ XXXX.....? SOUM
LABEL: UP TO 10 CHARS....? BUENAVISTA
TITLE: UP TO 35 CHARS....? MARY POPPINS
ARTIST: UP TO 25 CHARS...? SOUNDTRACK


FILE NO.....1001
CLASS.....8
ON TAPE.....1
TITLE KEY...MARY
ARTIST KEY..SOUM
LABEL.....BUENAVISTA
TITLE.....MARY POPPINS
ARTIST.....SOUNDTRACK


IF THIS IS INCORRECT HIT 'N', OTHERWISE JUST HIT 'RETURN'?


IF THERE IS NO MORE INPUT, AT THIS POINT
ENTER '0' WHEN ASKED FOR FILE NO.
FILE NO. @ XXXX.....? 1002
CLASSIFICATION @ X.....? 5
ON TAPE (1=YES) @ X.....? 1
OTHER @ XXX.....? 000
TITLE KEY @ XXXX.....? BEAT
ARTIST KEY @ XXXX.....? BEAT
LABEL: UP TO 10 CHARS....? CAPITOL
TITLE: UP TO 35 CHARS....? THE BEATLES AT HOLLYWOOD BOWL
ARTIST: UP TO 25 CHARS...? THE BEATLES


FILE NO.....1002
CLASS.....5
ON TAPE.....1
TITLE KEY...BEAT
ARTIST KEY..BEAT
LABEL.....CAPITOL
TITLE.....THE BEATLES AT HOLLYWOOD BOWL
ARTIST.....THE BEATLES


IF THIS IS INCORRECT HIT 'N', OTHERWISE JUST HIT 'RETURN'?

  RECORD NO.  3


IF THERE IS NO MORE INPUT, AT THIS POINT
ENTER '0' WHEN ASKED FOR FILE NO.
FILE NO. @ XXXX.....? 1003
CLASSIFICATION @ X.....? 6
ON TAPE (1=YES) @ X.....? 0
OTHER @ XXX.....? 000
TITLE KEY @ XXXX.....? KIKI
ARTIST KEY @ XXXX.....? DEEK
LABEL: UP TO 10 CHARS....? MCA
TITLE: UP TO 35 CHARS....? KIKI DEE
ARTIST: UP TO 25 CHARS...? KIKI DEE


FILE NO.....1003
CLASS.....6
ON TAPE.....0
TITLE KEY...KIKI
ARTIST KEY..DEEK
LABEL.....MCA
TITLE.....KIKI DEE
ARTIST.....KIKI DEE


IF THIS IS INCORRECT HIT 'N', OTHERWISE JUST HIT 'RETURN'?


IF THERE IS NO MORE INPUT, AT THIS POINT
ENTER '0' WHEN ASKED FOR FILE NO.
FILE NO. @ XXXX.....? 1004
CLASSIFICATION @ X.....? 6
ON TAPE (1=YES) @ X.....? 1
OTHER @ XXX.....? 000
TITLE KEY @ XXXX.....? MIRA
ARTIST KEY @ XXXX.....? IANM
LABEL: UP TO 10 CHARS....? COLUMBIA
```

```
TITLE: UP TO 35 CHARS....? MIRACLE ROW
ARTIST: UP TO 25 CHARS...? JANIS IAN

FILE NO.....1004
CLASS.....6
ON TAPE.....1
TITLE KEY...MIRA
ARTIST KEY..IANM
LABEL.....COLUMBIA
TITLE.....MIRACLE ROW
ARTIST.....JANIS IAN


IF THIS IS INCORRECT HIT 'N', OTHERWISE JUST HIT 'RETURN'?

  RECORD NO.  4

IF THERE IS NO MORE INPUT, AT THIS POINT
ENTER '0' WHEN ASKED FOR FILE NO.
FILE NO. @ XXXX.....? 0


IF THERE IS NO MORE INPUT, AT THIS POINT
ENTER '0' WHEN ASKED FOR FILE NO.
FILE NO. @ XXXX.....? 0

                  1 - INPUT NEW DATA
                  2 - EXAMINE/DELETE DISK FILES
                  3 - PRINT DISK FILES
                  4 - TRANSFER FILES
                  5 - INITIALIZE NEW FILE
                  8 - RETURN TO MAIN PGM.
                  9 - EXIT PROGRAM

                  WHICH OPTION? 3


ENTER FILE NAME TO BE PRINTED? ALBUMS
ENTER P OR T FOR PRINTER OR TERMINAL OUTPUT? T

THERE ARE  6  ALBUMS IN THE 'ALBUMS' FILE.

    RECORD NO. 2
 1001/8/1/000/MARY/SOUM/BUENAVISTA/MARY POPPINS/SOUNDTRACK
 1002/5/1/000/BEAT/BEAT/CAPITOL/THE BEATLES AT HOLLYWOOD BOWL/
       THE BEATLES

    RECORD NO. 3
 1003/6/0/000/KIKI/DEEK/MCA/KIKI DEE/KIKI DEE
 1004/6/1/000/MIRA/IANM/COLUMBIA/MIRACLE ROW/JANIS IAN

    RECORD NO. 4
 0
 0

ARE YOU READY? Y

                  1 - INPUT NEW DATA
                  2 - EXAMINE/DELETE DISK FILES
                  3 - PRINT DISK FILES
                  4 - TRANSFER FILES
                  5 - INITIALIZE NEW FILE
                  8 - RETURN TO MAIN PGM.
                  9 - EXIT PROGRAM

                  WHICH OPTION? 2


ENTER DISK FILE NAME TO BE MODIFIED OR DELETED? ALBUMS

THERE ARE  4  RECORDS IN ALBUMS FILE.


IF ALBUM 'A' OR 'B' IS TO BE CHANGED, ENTER '0'
OTHERWISE JUST HIT 'RETURN' WHEN ASKED 'NEW ALBUM'.


ENTER RECORD NO.? 1
NOT A VALID RECORD NO.
ENTER RECORD NO.? 2

ALBUM 'A': 1001/8/1/000/MARY/SOUM/BUENAVISTA/MARY POPPINS/SOUNDTRACK
ALBUM 'B': 1002/5/1/000/BEAT/BEAT/CAPITOL/THE BEATLES AT HOLLYWOOD
       BOWL/THE BE
ATLES

CHANGE? Y

NEW ALBUM 'A' ? 0
NEW ALBUM 'B' ?

ANYMORE? Y
ENTER RECORD NO.? 4

  ALBUM 'A': 0
  ALBUM 'B': 0

CHANGE? Y

NEW ALBUM 'A' ? 1005/8/1/000/STAR/SOUS/TWENTIETH/STAR WARS/SOUNDTRACK
NEW ALBUM 'B' ? 0

ANYMORE? N

                  1 - INPUT NEW DATA
                  2 - EXAMINE/DELETE DISK FILES
                  3 - PRINT DISK FILES
                  4 - TRANSFER FILES
                  5 - INITIALIZE NEW FILE
                  8 - RETURN TO MAIN PGM.
                  9 - EXIT PROGRAM

                  WHICH OPTION? 3


ENTER FILE NAME TO BE PRINTED? ALBUMS
ENTER P OR T FOR PRINTER OR TERMINAL OUTPUT? T

THERE ARE  6  ALBUMS IN THE 'ALBUMS' FILE.

    RECORD NO. 2
 0
 1002/5/1/000/BEAT/BEAT/CAPITOL/THE BEATLES AT HOLLYWOOD BOWL

    RECORD NO. 3
 1003/6/0/000/KIKI/DEEK/MCA/KIKI DEE/KIKI DEE
 1004/6/1/000/MIRA/IANM/COLUMBIA/MIRACLE ROW/JANIS IAN

    RECORD NO. 4
 1005/8/1/000/STAR/SOUS/TWENTIETH/STAR WARS/SOUNDTRACK
 0

ARE YOU READY? Y

                  1 - INPUT NEW DATA
                  2 - EXAMINE/DELETE DISK FILES
                  3 - PRINT DISK FILES
                  4 - TRANSFER FILES
                  5 - INITIALIZE NEW FILE
                  8 - RETURN TO MAIN PGM.
                  9 - EXIT PROGRAM

                  WHICH OPTION? 4


  ENTER NAME OF FILE TO BE TRANSFERED? ALBUMS

  ENTER NAME OF FILE TO BE CREATED? ALBUMS2

        4  DISK RECORDS IN THIS FILE

      NO. OF DISK RECORDS TO BE USED? 4

                  1 - TITLE SORT
                  2 - ARTIST SORT
                  3 - LABEL SORT
                  4 - FILE NO. SORT
                  5 - CLASS. SORT
                  ENTER # FOR SORT? 2

                  1 - INPUT NEW DATA
                  2 - EXAMINE/DELETE DISK FILES
                  3 - PRINT DISK FILES
                  4 - TRANSFER FILES
                  5 - INITIALIZE NEW FILE
                  8 - RETURN TO MAIN PGM.
                  9 - EXIT PROGRAM

                  WHICH OPTION? 3


  ENTER FILE NAME TO BE PRINTED? ALBUMS2
  ENTER P OR T FOR PRINTER OR TERMINAL OUTPUT? T

  THERE ARE  4  ALBUMS IN THE 'ALBUMS2' FILE.

     RECORD NO. 2
  1002/5/1/000/BEAT/BEAT/CAPITOL/THE BEATLES AT HOLLYWOOD BOWL/
      THE BEATLES
  1003/6/0/000/KIKI/DEEK/MCA/KIKI DEE/KIKI DEE

     RECORD NO. 3
  1004/6/1/000/MIRA/IANM/COLUMBIA/MIRACLE ROW/JANIS IAN
  1005/8/1/000/STAR/SOUS/TWENTIETH/STAR WARS/SOUNDTRACK

  ARE YOU READY? Y

                  1 - INPUT NEW DATA
                  2 - EXAMINE/DELETE DISK FILES
                  3 - PRINT DISK FILES
                  4 - TRANSFER FILES
                  5 - INITIALIZE NEW FILE
                  8 - RETURN TO MAIN PGM.
                  9 - EXIT PROGRAM

                  WHICH OPTION? 8


  SELECTION LIST FOR RECORD ALBUM PROGRAM

       1 - TITLE SORT AND PRINT
       2 - ARTIST SORT AND PRINT
       3 - LABEL SORT AND PRINT
       4 - FILE NO. SORT AND PRINT
       5 - CLASSIFICATION SORT AND PRINT
       6 - UTILITY: UPDATES, DELETES, CREATES FILES
       7 - EXIT

  ENTER THE TOTAL AMOUNT OF LISTINGS DESIRED? 5

       FOR LISTING NO. 1 YOUR CHOICE IS ? 1

       FOR LISTING NO. 2 YOUR CHOICE IS ? 2

       FOR LISTING NO. 3 YOUR CHOICE IS ? 3

       FOR LISTING NO. 4 YOUR CHOICE IS ? 4

       FOR LISTING NO. 5 YOUR CHOICE IS ? 5

  WHICH DISK FILE WILL BE USED IN THIS SORT? ALBUMX_S2

        3  DISK RECORDS IN THIS FILE

       NO. OF DISK RECORDS TO BE USED? 3

            * NOW SORTING 4 ALBUMS *


      THERE WILL BE APPROX.  20  PRINT LINES.

         THERE WILL BE APPROX.   5.0 PAGES.
```

Continued on following page

SAMPLE RUN CONTINUED

* * * * * ALBUM LISTING BY TITLE * * * * *

PAGE 1

TITLE	ARTIST	LABEL	CLASSIFICATION	FILE NO.	ON TAPE
THE BEATLES AT HOLLYWOOD BOWL	THE BEATLES	CAPITOL	OLDIES	01002	*
KIKI DEE	KIKI DEE	MCA	POPULAR	01003	
MIRACLE ROW	JANIS IAN	COLUMBIA	POPULAR	01004	*
STAR WARS	SOUNDTRACK	TWENTIETH	SOUNDTRACK	01005	*

* * * * * ALBUM LISTING BY ARTIST * * * * *

PAGE 2

ARTIST	TITLE	LABEL	CLASSIFICATION	FILE NO.	ON TAPE
THE BEATLES	THE BEATLES AT HOLLYWOOD BOWL	CAPITOL	OLDIES	01002	*
KIKI DEE	KIKI DEE	MCA	POPULAR	01003	
JANIS IAN	MIRACLE ROW	COLUMBIA	POPULAR	01004	*
SOUNDTRACK	STAR WARS	TWENTIETH	SOUNDTRACK	01005	*

* * * * * ALBUM LISTING BY LABEL * * * * *

PAGE 3

LABEL	ARTIST	TITLE	CLASSIFICATION	FILE NO.	ON TAPE
CAPITOL	THE BEATLES	THE BEATLES AT HOLLYWOOD BOWL	OLDIES	01002	*
COLUMBIA	JANIS IAN	MIRACLE ROW	POPULAR	01004	*
MCA	KIKI DEE	KIKI DEE	POPULAR	01003	
TWENTIETH	SOUNDTRACK	STAR WARS	SOUNDTRACK	01005	*

* * * * * ALBUM LISTING BY FILE NO. * * * * *

PAGE 4

FILE NO.	TITLE	ARTIST	LABEL	CLASSIFICATION	ON TAPE
01002	THE BEATLES AT HOLLYWOOD BOWL	THE BEATLES	CAPITOL	OLDIES	*
01003	KIKI DEE	KIKI DEE	MCA	POPULAR	
01004	MIRACLE ROW	JANIS IAN	COLUMBIA	POPULAR	*
01005	STAR WARS	SOUNDTRACK	TWENTIETH	SOUNDTRACK	*

* * * * * ALBUM LISTING BY CLASSIFICATION * * * * *

PAGE 5

CLASSIFICATION	TITLE	ARTIST	LABEL	FILE NO.	ON TAPE
OLDIES	THE BEATLES AT HOLLYWOOD BOWL	THE BEATLES	CAPITOL	01002	*
POPULAR	KIKI DEE	KIKI DEE	MCA	01003	
POPULAR	MIRACLE ROW	JANIS IAN	COLUMBIA	01004	*
SOUNDTRACK	STAR WARS	SOUNDTRACK	TWENTIETH	01005	*

More Than a Personal Computing Exposition, It's a Happening!

2nd ANNUAL
MIDWEST PERSONAL COMPUTING EXPOSITION

OCTOBER 5-8, 1978 • EXPOCENTER/Chicago (Adjacent To Merchandise Mart)

A week-long revel of special events, celebrity attractions, adventures into the future, surprises and assorted delights . . . **Now officially proclaimed, "Midwest Personal Computing Week" by Chicago's Mayor Bilandic.**

And highlighted by the second annual MIDWEST PERSONAL COMPUTING EXPO, which brings together the latest innovations in equipment, peripherals, accessories and software by such leading manufacturers, distributors and publications as:

Bits 'n Bytes Computer Stores
Compute-A-Bit
Digital Enterprises
Microcomp
Olson Electronics
Summagraphics
Telpar, Inc.
Artec Electronics
Cherry Electrical Products Corp.
Information Systems Inc.
Apple Computer Co.
Jade, Inc.
Hobbyworld
Radio Shack
Nabih's Inc.
Control Data Corp.
Chicago Computer Stores, Inc.
The Milwaukee Computer Store
Quantum Computer Works
Tek-Aids, Inc.
itty bitty Machine Co.
Compucolor Corp.
The Digital Group
Processor Technology
AAA Chicago Computer Center
Computer Electronic Modes, Inc.
Quay Corp.
Personal Computing
Minicomputer News
Chicagoland Business Services and Equipment News
Electronics Journal
Computer Dealer
Digital Design
Electronic Packaging and Production

Listing Complete As of May 1, 1978 . . . Many More Top-Name Exhibitors by October.

Special Emphasis on Meeting the Computer Needs of Small and Medium-Sized Businesses

Over 200 product displays anticipated by show time. 12,565 doctors, lawyers, business and computer professionals, educators, students, home users and hobbyists came last year. More than 20,000 expected in '78.

Easy access from all public transportation and expressways to Chicago's close-in Expocenter. Ample parking available.

Expanded seminar program for everyone — beginning or experienced. Details practical applications of today's personal computing technology, including:

- Introduction to BASIC
- How to Program Your Own Computer
- How to Use Resources of Million Dollar Computer Systems with Personal Computers
- Home Applications of Personal Computers — Tax Records, Environmental Control, etc.
- Computer Graphics for Business or Entertainment
- Time-sharing with a Microcomputer
- Plus many more.

Plan Now to Participate as Visitor or Exhibitor

The ONLY Personal Computing Show in Chicago Officially Sponsored by PERSONAL COMPUTING Magazine

Organized in 1978 by:

Industrial and Scientific Conference Management, Inc.
a respected leader in expositions and conferences, worldwide
222 W. Adams St., Chicago, IL 60606
(312) 263-4866

Return for Complete Information to:

MIDWEST PERSONAL COMPUTING EXPOSITION
c/o Industrial and Scientific Conference Management, Inc.
222 West Adams St., Chicago, IL 60606 • (312) 263-4866

☐ I am interested in ATTENDING. Please send registration information.

☐ I am interested in EXHIBITING. Please send contract and full details.

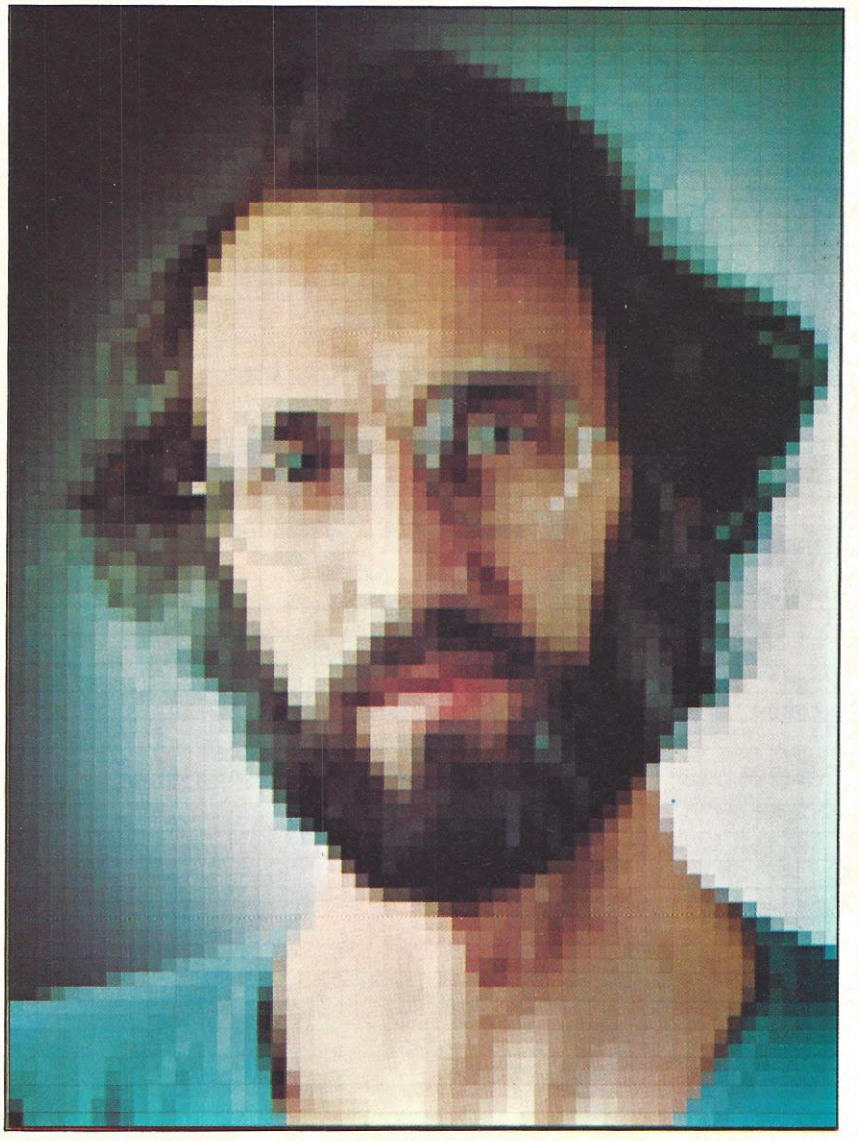
Name _____ Title _____

Company _____ Phone _____

Address _____

City _____ State _____ Zip _____

THE MAGIC OF A FACE



Spatially quantized transforms in full color, such as this one in a format of 60x80 (4800) squares, can be made instantly with an optical processor. An array of sensors matching the processing domains can be interfaced to a computer, eliminating the process of quantization by the computer, thus speeding up the whole operation and permitting the full capacity of the computer to be devoted to the problems of image analysis and recognition. (BLOCPIX® photo courtesy of Ed Manning, Watson-Manning, Inc., Stratford, Connecticut.)

BY PAUL SNIGIER

Can microcomputers remember and identify human faces accurately, rapidly and inexpensively? Well, if recent experiments conducted by Professor Leon Harmon of Case Western Reserve University are any indication, then you can expect this exciting possibility to become reality – perhaps within only one decade.

But why all this sudden interest? Such face-recognition systems, combined with voice-recognition units, could quickly identify people seeking access to classified documents or restricted areas. Utilized by airlines, or placed in public check points by law-enforcement agencies, such systems could screen passengers for known criminals or terrorists. And ultimately, banks and businesses could use them to augment credit cards and provide quick customer identification.

How did it all begin? While at Bell Labs, Professor Harmon began his initial experiments. Then, at Case Western he continued his research by first perfecting a "quantitative feature-description" approach. Human subjects were shown 256 frontal and profile photographs and described them by filling out questionnaires that listed 21 standard features. Nine features were selected for use in the computer – among them nose tip, chin, forehead, throat and mouth. Lines were then

drawn between these points to form various patterns. By analysis of angles and distances, a PDP-11/45 classified each face distinctly and uniquely.

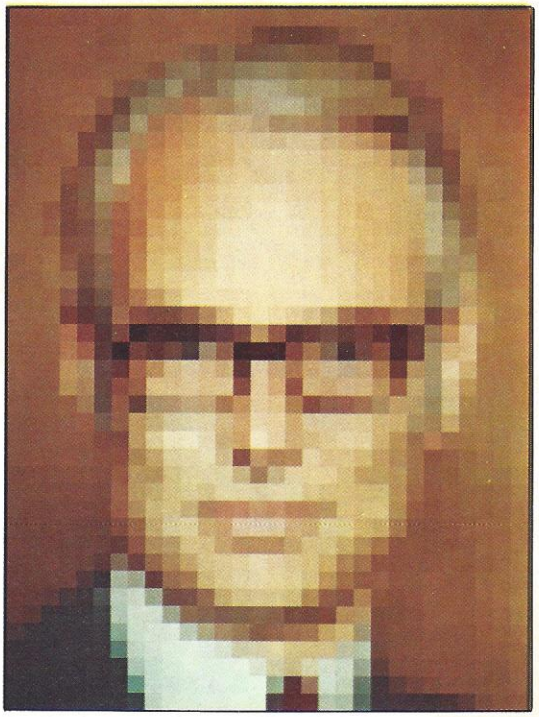
Using this information as a starting point, the researchers experimented with several face-recognition algorithms. Then, programming the minicomputer with these algorithms, they presented photographs at random to the computer. Instantly, it classified and sorted the data, storing the information for future use. Later, when "shown" these same photographs at random, the computer compared the unknown faces against those stored in its memory.

But difficulties arose. Although later face-recognition algorithms proved effective, the earlier versions left a bit to be desired. In these early algorithms, isolation was based upon a binary-decision approach. At each branching step in its search, the computer progressively discarded a number of faces by comparing the vector differences in a normalized Euclidian space to determine which members of the remaining subset it would retain. This sounded fine in theory, but in practice it proved ineffective; in fact, with just one error in the binary sequence, the correct choice was completely eliminated. Error rates reached 50 percent!

Professor Harmon soon adapted a more accurate process, rank-ordering, in which the computer assigned a goodness-of-fit measure or mathematical "weight" to each member of the population, that is, to all variables describing each face. At each identification step, the computer re-ranked population by weight. It repeated this process until the sequence converged to the correct selection, identifying the unknown face and also providing a numerical "confidence factor" that described how sure the computer was of its decision.

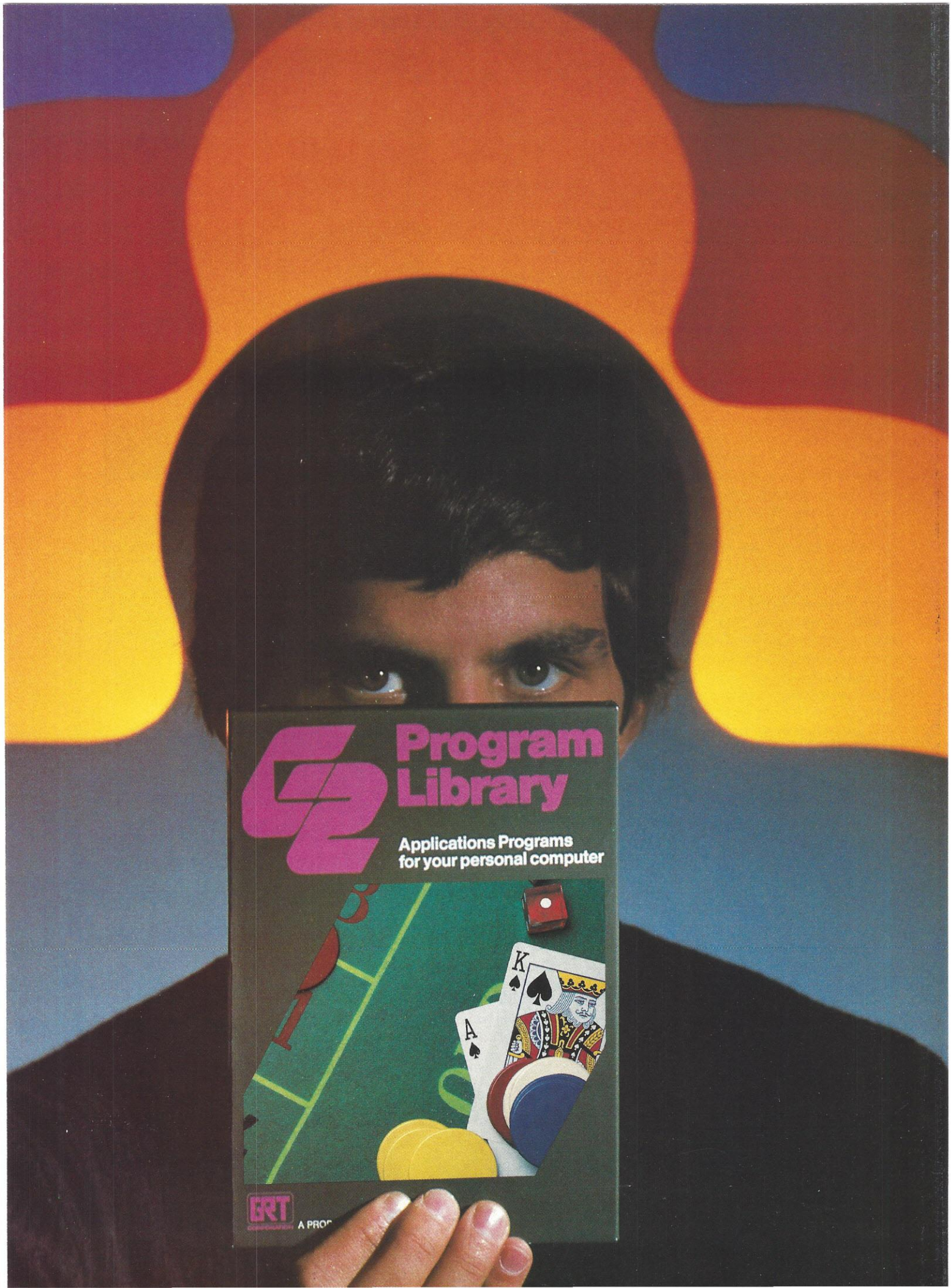
The researchers also tried another approach; trial-and-error experiments to determine minimal information required to pictorially represent a face. In this unusual procedure, the computer spatially quantitized the portrait into a matrix of 15-by-20 squares, digitized the analog output into six levels (four bits), and then integrated the brightness value corresponding to each of the original samples (a 25-by-25 "sub-matrix") lying within every square. When printed or viewed on a video terminal, these portraits resembled cubistic paintings; however, when viewed at a distance (30 to 40 picture diameters), the faces were easily recognizable by human observers. The researchers went one step further: they experimented with these quantitized portraits by smoothing them through computer low-pass filtering. This blurred the edges of each square by a local two-dimensional spatial-integration operation. The resulting image, unlike a cubistic image, resembled a ghostly face when seen closeup. However, the researchers rejected this approach and chose to work with the cubistic images.

Where will face-recognition by computer go from here? Although the researchers selected a homogenous sampling of individuals – male, beardless, short hair, no glasses, and with no obvious abnormal features such as scars – future face-recognition machines are expected to overlook these features (including plastic surgery) and identify such individuals by using ultrasonics or even x-rays to obtain skull portraits. □



Spatially quantized transform of fuli color input into square domains of equal size. Frequency of 30 x 40 (1200) domains approaches minimum level for easy human recognition of most faces. Machine recognition, employing the techniques similar to those explored in this article, may require fewer domains. (BLOCPIX® photo courtesy of Ed Manning, Watson-Manning, Inc., Stratford, Connecticut.)

Future face-recognition machines may overlook artificial features and identify people by ultrasonics or x-rays.


Program
Library
Applications Programs
for your personal computer
K
A
BRT

INTRODUCING THE G2™ PROGRAM LIBRARY.

Now you don't have to be a programmer to use your computer!

At last there's a reliable, professional source of useful and entertaining programs for your personal computer. The G2 Program Library from GRT unlocks the full power of your personal computer. Without any programming knowledge on your part.

The G2 Program Library consists of applications programs pre-recorded on cassette tapes. A detailed manual with each cassette provides complete instructions. And G2 programs contain prompting instructions. So they're easy to use every step of the way. Just load the tape and G2 puts your computer to work.

The first three G2 program tapes are available now: For exciting family games, get **Beat the House.** You'll find Blackjack, Craps, Roulette and Slot Machine programs, with truly random chances and realistic, casino-like payoffs.

Interested in your health? You'll want **Clinic.** Biorhythms, Dieting and Longevity programs give you new perspectives in health and happiness, along with some pleasant guidelines to follow.

Need a better handle on your money? Buy **Personal Finance.** Checkbook is a comprehensive program to balance your account and keep within your budget, while Best Choice is a novel way to make decisions based on the facts you know and the relative importance of each. Use it for everything from choosing a new car to selecting your next stereo.

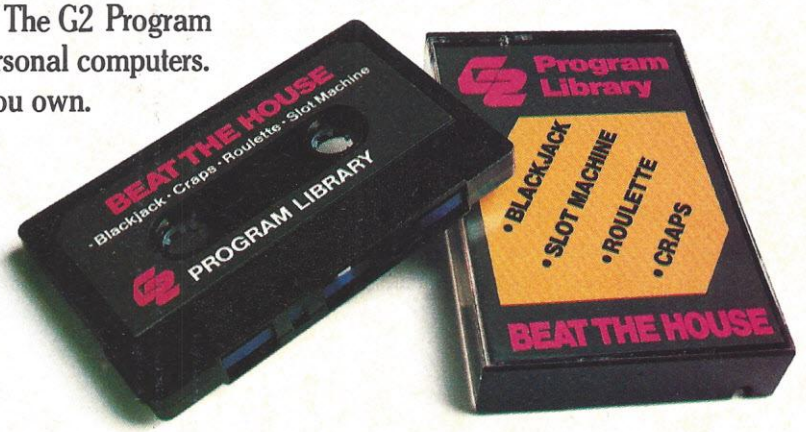
Soon, every dealer that handles personal computers will stock the G2 Program Library. For the G2 dealer nearest you, call us toll-free at 800/538-1770. (In California, 800/662-9810.) And if your favorite computer outlet doesn't yet offer G2, have him call the same number.

G2 is the newest product line from GRT, the world's largest independent producer of pre-recorded tapes. We've assembled a team of software specialists to produce the G2 Program Library. The finest library of personal computer programs available.

Soon you'll see G2 program tapes in many other fields of interest. Including advanced programs to challenge the interest of even the most serious computer user. The G2 Program Library is available for most popular brands of personal computers. Just select the package marked for the computer you own.

Start building your G2 Program Library now. And put the full power of your computer to work for you.

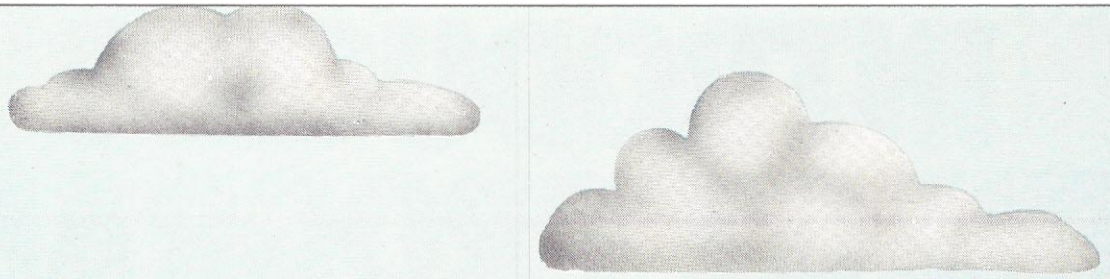
THE REASON YOU BOUGHT YOUR COMPUTER. G2™



GRT® CORPORATION

A product of GRT Corporation
Custom Products Division
1286 North Lawrence Station Road
Sunnyvale, California 94086

CIRCLE 11



BATTER-UP

BY JOE ROEHRIG

This article describes a simulated Major League baseball game using an 8080 microprocessor. The model is written in North Star BASIC release 3 and requires about 11.5K of free memory after BASIC is loaded. The model is a complete package and utilizes random access disk files to maintain records of the baseball game being simulated. Anyone familiar with BASIC should be able to tailor these programs to fit any particular hardware/software configuration. Those fortunate enough to be up and running with a North Star Disk System need only enter the listings.

For the moment, let's assume you're using a North Star System and the simulation programs have already been entered onto a diskette. Before loading BASIC, any new team-files desired should be initialized. Create a file name to correspond to the team name of every club being included in the simulation.

Each file will be 10 blocks of 256 bytes and will be assigned a type 3 (representing data) classification. In my sample runs, I use two team files, YANKS and REDS corresponding to the New York Yankees of the American League and the Cincinnati Reds of the National League.

To play the game, load program "BB" and type run. The model then lets you choose one of eight courses of action.

Figure 1 shows action 6 being selected. This completely clears and initializes a file for use in the simulation. In the sample, the "REDS" file is being initialized.

Throughout the sample runs, a reference number is printed after each operation and before branching back to the command mode. The model is composed of eight programs (one serves as a linking program) and this reference number corresponds to the free memory space after you executed each individual program. This reference is helpful in debugging, since you can utilize this pointer to determine which program is running when an error appears.

Inputting Statistics

Because the model is based on actual baseball statistics, you must supply data for individual players every time you construct a new team file. Action code 5 transfers control to the input mode.

Figure 1 shows data for the Reds being entered into the system. The user supplies the file name and selects one of two possible inputs. The choice hinges on the statistics available.

The Sporting News, a weekly newspaper selling for $1, is my usual source. During the baseball season, this publication prints weekly statistics covering all major league baseball players. A yearly summary by the paper (usually in print during January) contains a more detailed package than the weekly versions. Other sources of baseball statistics exist, such as team yearbooks, *The Sports Encyclopedia* and *Baseball.*

In the sample run, the final statistics mode is selected. Entered into the compuer, at the time, are the following data: the player's name; his stance (right, left or switch hitter); times at bat; hits; doubles; triples; home runs; walks and strike outs. Data on pitcher is entered separately and would include his name; how he throws (right or left); innings pitched; hits allowed; walks; strike outs and home runs allowed. If you select a non-final mode, walks and strike outs for each non-pitcher and home runs allowed by each pitcher would not be required. However, the simulation would be less accurate.

The sample shows how the program solicits the names of the players to be input. The file structure holds 18 non-pitchers, 12 pitchers and 3 possible starting line-ups. The user inputs only the amount of data he wishes. This enables the program to serve as a correction tool without revamping the entire file. In the sample, 9 non-pitchers, 3 pitchers and one line-up were supplied. The user, if he wishes can skip over sections by entering "0,0" at the desired input range.

Figure 2 shows a line-up entry. The simulation stores up to three lists thus preventing repetitous inputs. To enter a line-up, the player's ID number and position code are entered. The position codes are standard baseball scoring notations (2=catcher, 3=first base, 4=second base, 6=short stop, 5=third base, 7=left field, 8=center field, 9=right field). However, the game uses a designated hitter (code 0 here) making the position of the pitcher (code #1) invalid. The sample shows the program rejecting the pitcher being placed in batting position number 7.

What are player ID numbers?

Player ID numbers correspond to the input order. Action code 4 shows a complete listing of these names and ID numbers. In addition, the statistics and line-ups are also shown (see Figure 3). The sample lists the file called "YANKS", that was created earlier. Ratings for all statistics entered into the simulation have been calculated. These ratings are percentages and represent players' abilities to exceed the averages in the various categories.

Illustration by Stephen Fischer

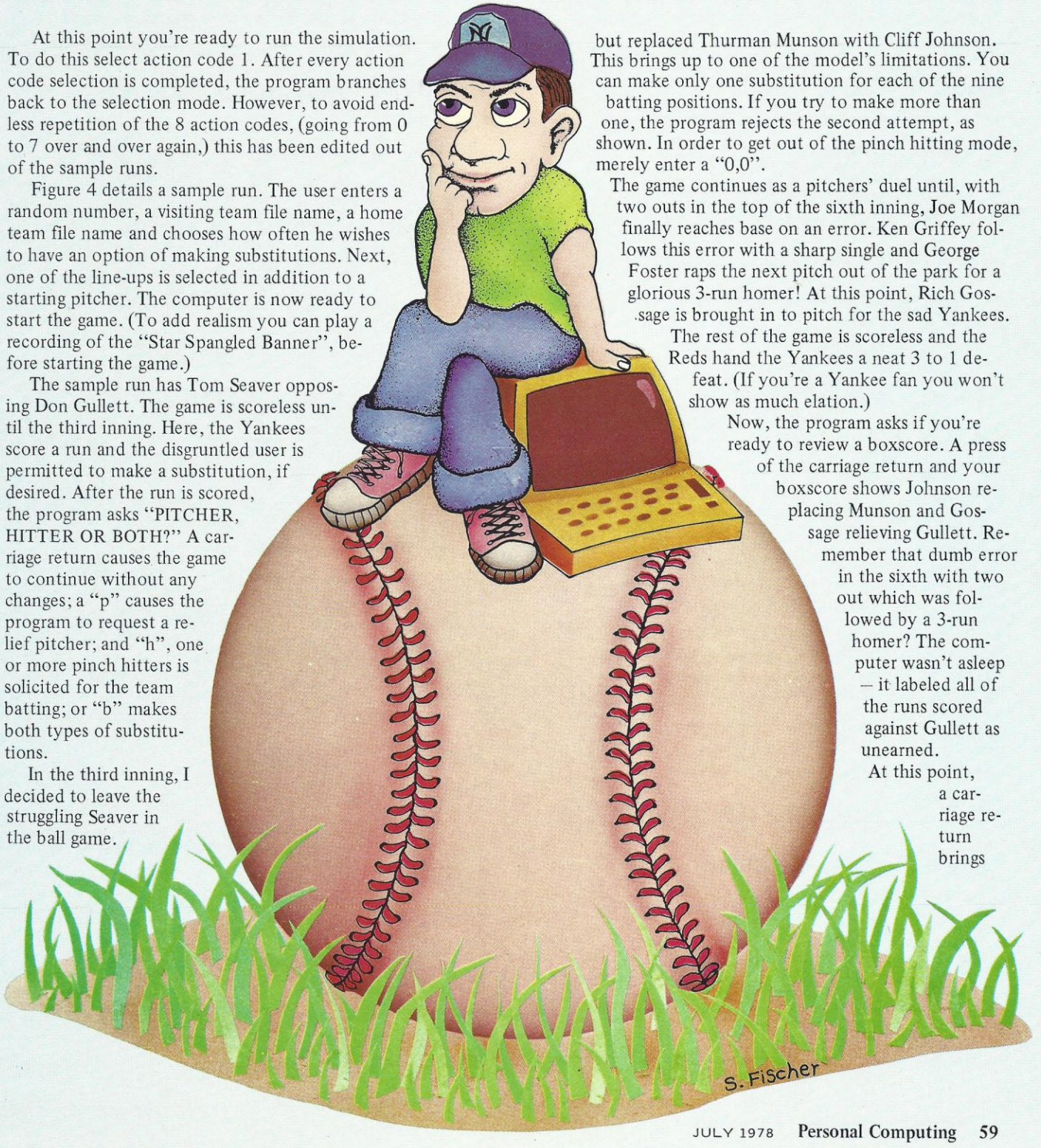
MICRO STLYE

At this point you're ready to run the simulation. To do this select action code 1. After every action code selection is completed, the program branches back to the selection mode. However, to avoid endless repetition of the 8 action codes, (going from 0 to 7 over and over again,) this has been edited out of the sample runs.

Figure 4 details a sample run. The user enters a random number, a visiting team file name, a home team file name and chooses how often he wishes to have an option of making substitutions. Next, one of the line-ups is selected in addition to a starting pitcher. The computer is now ready to start the game. (To add realism you can play a recording of the "Star Spangled Banner", before starting the game.)

The sample run has Tom Seaver opposing Don Gullett. The game is scoreless until the third inning. Here, the Yankees score a run and the disgruntled user is permitted to make a substitution, if desired. After the run is scored, the program asks "PITCHER, HITTER OR BOTH?" A carriage return causes the game to continue without any changes; a "p" causes the program to request a relief pitcher; and "h", one or more pinch hitters is solicited for the team batting; or "b" makes both types of substitutions.

In the third inning, I decided to leave the struggling Seaver in the ball game.



but replaced Thurman Munson with Cliff Johnson. This brings up to one of the model's limitations. You can make only one substitution for each of the nine batting positions. If you try to make more than one, the program rejects the second attempt, as shown. In order to get out of the pinch hitting mode, merely enter a "0,0".

The game continues as a pitchers' duel until, with two outs in the top of the sixth inning, Joe Morgan finally reaches base on an error. Ken Griffey follows this error with a sharp single and George Foster raps the next pitch out of the park for a glorious 3-run homer! At this point, Rich Gossage is brought in to pitch for the sad Yankees.

The rest of the game is scoreless and the Reds hand the Yankees a neat 3 to 1 defeat. (If you're a Yankee fan you won't show as much elation.)

Now, the program asks if you're ready to review a boxscore. A press of the carriage return and your boxscore shows Johnson replacing Munson and Gossage relieving Gullett. Remember that dumb error in the sixth with two out which was followed by a 3-run homer? The computer wasn't asleep – it labeled all of the runs scored against Gullett as unearned.

At this point, a carriage return brings

you back to the action selection mode. Any string of characters causes another game to be played between the same two teams (the simulation can have numerous team files, but to select different teams you must return to the action selection mode). If another game is desired, the program allows you to select a new line-up and a fresh starting pitcher. In the sample, another game was played and the Yankees dropped this one also, 10 to 7.

How About Statistics

An action code selection of 2, displays team statistics as depicted in Figure 6. Here you are required to supply the file name only. The statistics are fairly complete and even include pitchers' wins and losses (a 4-hour bug was incurred in developing the boxscores program by the way). But we can't leave you with this bare minimum of data. League leaders and standings have to be extracted from the data.

Action code 3 produces simulated historical data as shown in Figure 7. However, before this can be generated, you must tell the computer what files are in the League. As illustrated, the number of teams is input first

Batting averages are naturally high for the leaders because only two games have been played. In total, the Reds' overall average of .270 and the Yankees' .254 appears to be realistic. To end the simulation, enter a "0" action code.

The Programs

There are eight programs in all and the coding totals 334 multi-statement lines. Figure 8 is a listing of the North Star directory. The number of lines per program has been added to this listing. As mentioned earlier, one program, "BB", does nothing more than link the other seven programs together utilizing North Star's chain command. If desired, this program can be eliminated. This decision would also get rid of the action codes. Instead of having action codes, you would have to load the seven programs as desired. □

Figure 1 - Reds Data

```
THIS IS THE BASEBALL SIMULATION.  THE COMMANDS ARE:
       0  TO END
       1  TO PLAY A GAME
       2  TO DISPLAY TEAM STATISTICS
       3  LEAGUE STANDINGS DISPLAY
       4  LIST TEAM ROSTER
       5  INPUT OR CORRECT A ROSTER
       6  COMPLETELY CLEAR A FILE
       7  CLEAR STATISTICS ONLY FROM A FILE
       COMMAND NUMBER ? 6

FILE TO BE CLEARED: REDS
IF YOU ENTER A RETURN FILE REDS WILL BE COMPLETELY ERASED ?
 19160

THIS IS THE BASEBALL SIMULATION.  THE COMMANDS ARE:
       0  TO END
       1  TO PLAY A GAME
       2  TO DISPLAY TEAM STATISTICS
       3  LEAGUE STANDINGS DISPLAY
       4  LIST TEAM ROSTER
       5  INPUT OR CORRECT A ROSTER
       6  COMPLETELY CLEAR A FILE
       7  CLEAR STATISTICS ONLY FROM A FILE
       COMMAND NUMBER ? 5

FILE: REDS
0 FOR WEEKLY TYPE STATS OR 1 FOR FINAL STATS? 1
# FROM AND TO FOR NON-PITCHERS? 1,9
BATS: 0-RIGHT  1-LEFT  2-BOTH
NAME? FOSTER
BATS, AB, H, D, T, HR, W, K ? 0,615,197,31,2,52,61,107
NAME? GRIFFEY
BATS, AB, H, D, T, HR, W, K ? 1,585,186,35,8,12,69,84
NAME? ROSE
BATS, AB, H, D, T, HR, W, K ? 2,655,204,38,7,9,66,42
NAME? DRIESSEN
BATS, AB, H, D, T, HR, W, K ? 1,536,161,31,4,17,64,85
NAME? MORGAN
BATS, AB, H, D, T, HR, W, K ? 1,521,150,21,6,22,137,58
NAME? BENCH
BATS, AB, H, D, T, HR, W, K ? 0,494,136,34,2,31,58,95
NAME? CONCEPCION
BATS, AB, H, D, T, HR, W, K ? 0,572,155,26,3,8,46,77
NAME? GERONIMO
BATS, AB, H, D, T, HR, W, K ? 1,492,131,22,4,10,35,89
NAME? BAILEY
BATS, AB, H, D, T, HR, W, K ? 0,79,20,2,1,2,12,10
# FROM AND TO FOR PITCHERS? 1,3
THROWS: 0-RIGHT 1-LEFT
NAME? SEAVER
THROWS, IP, H, HR, W, K ? 0,261,199,19,66,196
NAME? BORBON
THROWS, IP, H, HR, W, K ? 0,127,131,7,24,48
NAME? NORMAN
THROWS, IP, H, HR, W, K ? 1,221,200,28,98,160

INPUT LINE-UPS
THERE ARE 3 LINE-UPS, WHICH ONE IS TO BE UPDATED OR 0 TO END ? 1
```

Figure 2 - Line-up Entry

```
INPUT PLAYER ID#, POS #
?3,5
?5,4
?2,9
?1,7
?6,2
?4,3
?9,1
INVALID ENTRY, REDO BATTING POSITION 7
?9,0
?7,6
?8,8

INPUT LINE-UPS
THERE ARE 3 LINE-UPS, WHICH ONE IS TO BE UPDATED OR 0 TO END ? 0
```

Figure 3 - Statistics & Line-up

```
FILE: YANKS

 # NAME        B      H      D      T      HR      W      K
== ========== = ====== ====== ====== ====== ====== ======
 1 PINIELLA.. 0  .066   .190   .220   .011  -.032  -.068
 2 RIVERS.... 1  .061   .105   .134  -.031  -.056  -.087
 3 ALSTON.... 1  .061   .333   .333  -.019  -.017   .128
 4 ZEBER..... 2  .059   .167   .167   .047   .034   .045
 5 MUNSON.... 0  .043   .170   .200   .003  -.026  -.072
 6 JOHNSON... 1  .033   .254   .254   .163   .043   .009
 7 CHAMBLISS. 1  .022   .206   .245   .003  -.017  -.034
 8 JACKSON... 1  .022   .331   .347   .118   .038   .139
 9 RANDOLPH.. 1  .010   .190   .265  -.069   .017  -.073
10 WHITE..... 2  .004   .200   .216   .005   .038  -.052
11 KLUTTS.... 0  .002   .333   .333   .154   .030  -.114
12 BLAIR..... 0 -.002   .103   .179  -.003  -.035  -.073
13 STANLEY... 0 -.003   .000   .000  -.012   .061  -.029
14 NETTLES... 1 -.010   .204   .239   .151   .016  -.025
15 DENT...... 0 -.017   .164   .200  -.028  -.012  -.127
16 HEALY..... 0 -.040   .333   .333  -.096  -.005   .045
17 SPENCER... 1 -.017   .163   .173   .059  -.016  -.064
18 ..........  0  .000   .000   .000   .000   .000   .000

 # NAME        T      H      W      K      HR
== ========== = ====== ====== ====== ======
 1 LYLE...... 1 -.010  -.027  -.028  -.035
 2 GUIDRY.... 1 -.037  -.008   .092  -.027
 3 TIDROW.... 0 -.012  -.020  -.009   .037
 4 FIGUEROA.. 0 -.011  -.010  -.050  -.012
 5 GULLETT... 1 -.028   .019   .056   .006
 6 CLAY...... 0 -.012   .015  -.078   .017
 7 HUNTER.... 1 -.010  -.007  -.076   .116
 8 PATTERSON. 0  .027   .046   .108  -.017
 9 HOLTZMAN.. 1  .078  -.015  -.136  -.029
10 EASTWICK.. 0  .031  -.017  -.032  -.017
11 GOSSAGE... 0 -.091   .011   .199   .020
12 MESSERSMIT 0 -.003   .004   .036   .023

LINE-UP NUMBER 1    LINE-UP NUMBER 2    LINE-UP NUMBER 3
================    ================    ================
RANDOLPH..  2B      RANDOLPH..  2B      BLAIR.....  CF
RIVERS....  CF      RIVERS....  CF      ALSTON....  RF
MUNSON....   C      MUNSON....  DH      WHITE.....  LF
JACKSON...  RF      PINIELLA..  LF      JACKSON...  DH
PINIELLA..  LF      JOHNSON...   C      SPENCER...  1B
SPENCER...  DH      JACKSON...  RF      HEALY.....   C
NETTLES...  3B      NETTLES...  3B      KLUTTS....  3B
CHAMBLISS.  1B      CHAMBLISS.  1B      ZEBER.....  2B
DENT......  SS      DENT......  SS      STANLEY...  SS
```

Figure 4 - Sample Run

```
RANDOM # ? 8
VISITING TEAM? REDS
HOME TEAM? YANKS
0 FOR SUB AFTER EVERY AT BAT OR 1 FOR AFTER A RUN? 1
INPUT VISITING TEAM'S LINE-UP #(1-3), STARTING PITCHER(1-12)? 1,1
INPUT HOME TEAM'S LINE-UP #(1-3), STARTING PITCHER(1-12)? 1,5
INNING # 1
==========
VISITING TEAM BATTING
ROSE..... IS RETIRED
MORGAN.... IS RETIRED
GRIFFEY... IS RETIRED

HOME TEAM BATTING
RANDOLPH.. IS RETIRED
RIVERS.... GETS A 2 BASE HIT                RUNNER ON SECOND
MUNSON.... STRIKES OUT
JACKSON... STRIKES OUT

SCORE AFTER 1 INNINGS REDS       0  YANKS      0

INNING # 2
==========
VISITING TEAM BATTING
```

```
FOSTER.... IS RETIRED
BENCH..... IS RETIRED
DRIESSEN.. IS RETIRED

HOME TEAM BATTING
PINIELLA.. GETS A 2 BASE HIT              RUNNER ON SECOND
SPENCER... STRIKES OUT
NETTLES... STRIKES OUT
CHAMBLISS. STRIKES OUT

SCORE AFTER 2 INNINGS REDS       0  YANKS      0


INNING # 3
==========
VISITING TEAM BATTING
BAILEY.... IS RETIRED
CONCEPCION IS RETIRED
GERONIMO.. IS RETIRED

HOME TEAM BATTING
DENT...... STRIKES OUT
RANDOLPH.. STRIKES OUT
RIVERS.... GETS A 1 BASE HIT              RUNNER ON FIRST
MUNSON.... GETS A 1 BASE HIT              RUNNER ON FIRST SECOND
JACKSON... WALKS                          RUNNER ON FIRST SECOND THIRD
PINIELLA.. GETS A 1 BASE HIT              RUNNER ON FIRST SECOND THIRD
                                           1 RUNS SCORE

PITCHER, HITTER OR BOTH? H
BATTING ORDER POSITION, ID#? 3,6
SUBSTUTITION MADE
BATTING ORDER POSITION, ID#? 3,16
YOU ALREADY MADE A SUB IN THAT POSITION
BATTING ORDER POSITION, ID#? 0,0
NO CHANGE
SPENCER... IS RETIRED

SCORE AFTER 3 INNINGS REDS       0  YANKS      1


INNING # 4
==========
VISITING TEAM BATTING
ROSE...... GETS A 1 BASE HIT              RUNNER ON FIRST
MORGAN.... IS RETIRED
GRIFFEY... WALKS                          RUNNER ON FIRST  SECOND
FOSTER.... IS RETIRED                     RUNNER ON SECOND THIRD
BENCH..... IS RETIRED

HOME TEAM BATTING
NETTLES... WALKS                          RUNNER ON FIRST
CHAMBLISS. IS RETIRED
DENT...... IS RETIRED
RANDOLPH.. IS RETIRED

SCORE AFTER 4 INNINGS REDS       0  YANKS      1

INNING # 5
==========
VISITING TEAM BATTING
DRIESSEN.. STRIKES OUT
BAILEY.... IS RETIRED
CONCEPCION IS RETIRED

HOME TEAM BATTING
RIVERS.... IS RETIRED
JOHNSON... IS RETIRED
JACKSON... STRIKES OUT

SCORE AFTER 5 INNINGS REDS       0  YANKS      1

INNING # 6
==========
VISITING TEAM BATTING
GERONIMO.. STRIKES OUT
ROSE...... IS RETIRED
MORGAN.... REACHES ON AN ERROR            RUNNER ON FIRST
GRIFFEY... GETS A 1 BASE HIT              RUNNER ON FIRST  THIRD
FOSTER.... HITS A HOME RUN
                                           3 RUNS SCORE

PITCHER, HITTER OR BOTH? P
ID# OF NEW PITCHER? 11
RELIEF PITCHER IN
BENCH..... IS RETIRED

HOME TEAM BATTING
PINIELLA.. GETS A 1 BASE HIT              RUNNER ON FIRST
SPENCER... IS RETIRED
NETTLES... WALKS                          RUNNER ON FIRST  SECOND
CHAMBLISS. IS RETIRED                     RUNNER ON SECOND THIRD
DENT...... IS RETIRED

SCORE AFTER 6 INNINGS REDS       3  YANKS      1

INNING # 7
==========
VISITING TEAM BATTING
DRIESSEN.. IS RETIRED
BAILEY.... GETS A 1 BASE HIT              RUNNER ON FIRST
CONCEPCION IS RETIRED
GERONIMO.. STRIKES OUT

HOME TEAM BATTING
RANDOLPH.. WALKS                          RUNNER ON FIRST
RIVERS.... IS RETIRED
JOHNSON... STRIKES OUT
JACKSON... GETS A 1 BASE HIT              RUNNER ON FIRST  THIRD
PINIELLA.. IS RETIRED

SCORE AFTER 7 INNINGS REDS       3  YANKS      1

INNING # 8
==========
VISITING TEAM BATTING
ROSE...... IS RETIRED
MORGAN.... WALKS                          RUNNER ON FIRST
GRIFFEY... STRIKES OUT
FOSTER.... IS RETIRED

HOME TEAM BATTING
SPENCER... IS RETIRED
NETTLES... STRIKES OUT
CHAMBLISS. IS RETIRED

SCORE AFTER 8 INNINGS REDS       3  YANKS      1

INNING # 9
==========
VISITING TEAM BATTING
BENCH..... STRIKES OUT
DRIESSEN.. GETS A 2 BASE HIT              RUNNER ON SECOND
BAILEY.... GETS A 1 BASE HIT              RUNNER ON FIRST  THIRD
CONCEPCION IS RETIRED
GERONIMO.. STRIKES OUT

HOME TEAM BATTING
DENT...... GETS A 1 BASE HIT              RUNNER ON FIRST
RANDOLPH..
 GROUNDS INTO A DOUBLE PLAY
RIVERS.... GETS A 1 BASE HIT              RUNNER ON FIRST
JOHNSON... IS RETIRED

SCORE AFTER 9 INNINGS REDS       3  YANKS      1
```

	1	2	3	4	5	6	7	8	9	TOT
REDS	0	0	0	0	0	3	0	0	0	3
YANKS	0	0	1	0	0	0	0	0	0	1

Figure 5 - Box Score

```
BOXSCORE  REDS                                    YANKS
```

NAME	POS	AB	HITS	HR	RBI	NAME	POS	AB	HITS	HR	RBI
ROSE.....	3B	4	1	0	0	RANDOLPH..	2B	4	0	0	0
MORGAN....	2B	3	0	0	0	RIVERS....	CF	5	3	0	0
GRIFFEY...	RF	3	1	0	0	MUNSON....	C	2	1	0	0
						JOHNSON...		3	0	0	0
FOSTER....	LF	4	1	1	3	JACKSON...	RF	3	1	0	0
BENCH.....	C	4	0	0	0	PINIELLA..	LF	4	3	0	1
DRIESSEN..	1B	4	1	0	0	SPENCER...	DH	4	0	0	0
BAILEY....	DH	4	2	0	0	NETTLES...	3B	2	0	0	0
CONCEPCION	SS	4	0	0	0	CHAMBLISS.	1B	4	0	0	0
GERONIMO..	CF	4	0	0	0	DENT.....	SS	4	1	0	0
		==	==	=	==			==	==	=	==
TOTAL		34	6	1	3	TOTAL		35	9	0	1

```
READY FOR PITCHING STATS?
PITCHING
```

NAME	IP	H	R	ER	W	K	DECISION
SEAVER....	9.0	9	1	1	4	10	WINNER
GULLETT...	5.7	3	3	0	1	2	LOSSER
GOSSAGE...	3.3	3	0	0	1	4	

```
RETURN TO END OR ANYTHING ELSE FOR ANOTHER GAME? PLAY
```

Figure 6 - Team Statistics

```
WHAT FILE? YANKS
```

NAME	AT BATS	HITS	H.R.	RBI'S	AVE
PINIELLA..	4	3	0	1	.750
RIVERS....	5	3	0	0	.600
ALSTON....	5	0	0	0	.000
ZEBER.....	3	2	0	1	.667
MUNSON....	2	1	0	0	.500
JOHNSON...	3	0	0	0	.000
CHAMBLISS.	4	0	0	0	.000
JACKSON...	7	1	0	0	.143
RANDOLPH..	4	0	0	0	.000
WHITE.....	5	2	0	0	.400
KLUTTS....	3	2	0	0	.667
BLAIR.....	5	1	0	2	.200
STANLEY...	3	1	1	2	.333
NETTLES...	2	0	0	0	.000
DENT......	4	1	0	0	.250
HEALY.....	3	0	0	0	.000
SPENCER...	9	1	1	2	.111
.....	0	0	0	0	.000
TOTALS....	71	18	2	8	.254

```
READY FOR PITCHERS?
```

PITCHERS	I.P.	HITS	RUNS	E.R.	WALKS	KO'S	WON	LOST	PCT	ERA
LYLE.....	2.0	4	4	3	2	2	0	0	.000	13.50
GUIDRY....	5.0	7	3	3	2	3	0	0	.000	5.40
TIDROW....	.7	2	3	0	1	0	0	1	.000	.00
FIGUEROA..	.0	0	0	0	0	0	0	0	.000	.00
GULLETT...	5.7	3	3	0	1	2	0	1	.000	.00
CLAY.....	.0	0	0	0	0	0	0	0	.000	.00
HUNTER....	.0	0	0	0	0	0	0	0	.000	.00
PATTERSON.	.0	0	0	0	0	0	0	0	.000	.00
HOLTZMAN..	.0	0	0	0	0	0	0	0	.000	.00
EASTWICK..	1.3	1	0	0	0	1	0	0	.000	.00
GOSSAGE...	3.3	3	0	0	1	4	0	0	.000	.00
MESSERSMIT	.0	0	0	0	0	0	0	0	.000	.00
TOTALS....	18.0	20	13	6	7	12	0	2	.000	3.00

Figure 7 - Historical Data

```
ENTER 0 TO SET OR CHANGE LEAGUE FILE ? 0
HOW MANY TEAMS IN THE LEAGUE ? 2
GIVE THE FILE NAME FOR THE 2 TEAMS IN THE LEAGUE
  1  ? YANKS
  2  ? REDS
```

```
LEAGUE LEADERS
==============
NAME          AVE  NAME          HR  NAME          RBI  NAME          HITS
========== ====  ========== ====  ========== ====  ========== ====
PINIELLA... .750  FOSTER....    2  FOSTER....    5  PINIELLA..    3
ZEBER..... .667  STANLEY...    1  GERONIMO..    2  RIVERS....    3
KLUTTS.... .667  SPENCER...    1  BLAIR.....    2  FOSTER....    3
RIVERS.... .600  GERONIMO..    1  STANLEY...    2  GRIFFEY...    3
MUNSON.... .500  ----------    0  SPENCER...    2  GERONIMO..    3

LEAGUE LEADERS
==============
NAME          PCT  NAME          KO  NAME          WINS  NAME          ERA
========== ====  ========== ====  ========== ====  ========== ======
SEAVER....1.000  SEAVER....   10  SEAVER....    1  TIDROW....   .00
BORBON....1.000  NORMAN....    6  BORBON....    1  GULLETT...   .00
LYLE..... .000  GOSSAGE...    4  LYLE.....     0  EASTWICK..   .00
GUIDRY.... .000  GUIDRY....    3  GUIDRY....    0  GOSSAGE...   .00
TIDROW.... .000  LYLE.....     2  TIDROW....    0  SEAVER....  1.00

READY FOR STANDINGS ?

STANDINGS
=========
TEAM        WINS LOST   PCT    GB
========== ==== ==== ===== =====
REDS------     2    0 1.000    .0
YANKS-----     0    2  .000   2.0

FILE TO BE CLEARED: REDS
IF YOU ENTER A RETURN FILE REDS WILL HAVE ITS STATISTICS REMOVED?
 19166

THIS IS THE BASEBALL SIMULATION.  THE COMMANDS ARE:
      0  TO END
      1  TO PLAY A GAME
      2  TO DISPLAY TEAM STATISTICS
      3  LEAGUE STANDINGS DISPLAY
      4  LIST TEAM ROSTER
      5  INPUT OR CORRECT A ROSTER
      6  COMPLETELY CLEAR A FILE
      7  CLEAR STATISTICS ONLY FROM A FILE
      COMMAND NUMBER ? 0
```

Figure 8 - North Star Directory

```
BB          4   4   2
BB-GAME     8  26   2
BB-INPUT   34   6   2
BB-ROST    40   4   2
BB-S-F     44   2   2
BB-C-F     46   2   2
BB-STAT    48   8   2
BB-STAND   56  12   2
BBLEAG     68   1   3
YANKS      69  10   3
REDS       79  10   3
```

This file shows file name, block address of the file, file size in 256 byte blocks and type (2= program or 3=date).

BB - Program Listing

```
5 INPUT"RANDOM # ? ";A\IFA<0THENA=0-A\IFA=0THENA=.1
7 A=A*.1\IFA>=1THEN7\B=RND(A)\FORB=1TO5\C=RND(0)\NEXT
10 DIMT$(20),H(1,17,10),P(1,11,12),N$(600),L(1,2,8,1),G(1,18,3),T(1,4)
12 DIMS(1,11,6),P$(20),W(1,17),B(7),R(1,9),E(7),A(1,2),B$(18)
14 X=13757/157731\Y=3644/38037\Z=38037/143974\W=21722/105937
18 P$="DH C1B2B3BSSLFCFRF P"\LINE80\B$="FIRST SECONDTHIRD "
20 INPUT"VISITING TEAM? ",T$(1,10)\OPEN#0,T$(1,10)
30 INPUT"HOME TEAM? ",T$(11,20)\OPEN#1,T$(11,20)
35 INPUT"0 FOR SUB AFTER EVERY AT BAT OR 1 FOR AFTER A RUN? ",P
40 FORA=0TO1\B=A*300\FORC=0TO17\D=B+1+(C*10)\E=D+9
50 READ#A,N$(D,E),H(A,C,0),F,H(A,C,1),H(A,C,2),H(A,C,3),H(A,C,4),
        H(A,C,5)
60 READ#A,H(A,C,6),H(A,C,7),H(A,C,8),H(A,C,9),H(A,C,10)\NEXT
70 FORC=0TO11\D=B+181+(C*10)\E=D+9
80 READ#A,N$(D,E),P(A,C,0),F,P(A,C,1),P(A,C,2),P(A,C,3),P(A,C,4),
        P(A,C,5)
90 READ#A,P(A,C,6),P(A,C,7),P(A,C,8),P(A,C,9),P(A,C,10),P(A,C,11),
        P(A,C,12)
100 NEXT\FORC=0TO2\FORD=0TO8\READ#A,L(A,C,D,0),L(A,C,D,1)
        \NEXT\NEXT\NEXT
110 FORA=0TO1\IFA>0THEN120\!"INPUT VISITING TEAM'S ",\GOTO130
120 !"INPUT HOME TEAM'S ",
130 INPUT"LINE-UP #(1-3), STARTING PITCHER(1-12)? ",B,C\A(A,2)=B-1
135 FORF=0TO17\W(A,F)=-1\NEXT
140 IFB<10RB>3THEN130\IFC<10RC>12THEN130\B=B-1\C=C-1\IFA<>0THEN145\D=
        B\E=C
145 FORF=0TO16STEP2\W(A,F)=L(A,B,F/2,0)\NEXT\A(0,1)=C\A(1,1)=E
150 FORF=0TO4\T(A,F)=P(A,C,F)\NEXT\NEXT
160 T=9\S1=0\S2=0\A(0,0)=0\A(1,0)=0
200 FORG=1TOT\!"INNING #",%2I,G\!"==========="\!"VISITING TEAM BATTING"
205 FORA=0TO1\J=A(A,1)\R=0
206 IFG=TANDA=1ANDR(1,0)>R(0,0)THENEXIT596
210 IFR>0THEN!TAB(40),R," RUNS SCORE"\IFP>0ANDR>0THENGOSUB950\IFP=
        0THENGOSUB950
211 IFG=TANDA=1ANDR(1,0)>R(0,0)THENEXIT596\R=0
212 I=A(A,0)\A(A,0)=A(A,0)+1\IFA(A,0)>8THENA(A,0)=0
215 L=0\N=0\M=W(A,2*I)\IFW(A,(2*I)+1)>0THENM=W(A,(2*I)+1)\M=
        M-1\GOSUB850
220 IF(T(A,2)+X+H(A,M,5))>RND(0)THEN400
222 G(A,M,0)=G(A,M,0)+1
225 K=0\IFT(A,0)=1THEN250
230 IFH(A,M,0)=0THENK=-.008\GOTO270
250 IFH(A,M,0)=1THENK=-.016
270 IFRND(0)>(Z+K+T(A,1)+H(A,M,1))THEN500
275 G(A,M,1)=G(A,M,1)+1\S(A,J,1)=S(A,J,1)+1
277IF((3*S(A,J,1))-3)<S(A,J,0)THEN280\T(A,1)=T(A,1)+.002\T(A,2)=
        T(A,2)+.002
280 IFRND(0)>(Y+K+T(A,4)+H(A,M,4))THEN300
285 !" HITS A HOME RUN  "
290 Q=4\N=4\GOSUB700\G(A,M,2)=G(A,M,2)+1\GOSUB800\GOTO210
300 P=RND(0)\N=1\IFH(A,M,3)>PTHENN=3
310 IFH(A,M,2)>PTHENN=2\Q=N
320 IFRND(0)>.5THENQ=Q+1
330 GOSUB700\GOSUB800\!" GETS A",N," BASE HIT",
340 GOSUB860\GOTO210
400 !" WALKS ",N=1\Q=1\GOSUB650\GOSUB860\N=1\GOSUB900
405 S(A,J,4)=S(A,J,4)+1\GOTO210
500 O1=O1+1\O2=O2+1
505 P=1\IF(T(A,3)+H(A,M,6)+W)<RND(0)THEN520
510 !" STRIKES OUT"\S(A,J,5)=S(A,J,5)+1\GOTO582
520 IF.04<RND(0)THEN530\Q=1\N=1\GOSUB700
525!" REACHES ON AN ERROR",\O1=O1-1\GOSUB860\GOTO210
530 IFB(1)=0THEN550
535 IFO1>1THEN550\IFRND(0)>.1THEN550\P=2
538 B(1)=0\E(1)=0
540 N=0\Q=1\O1=O1+1\O2=O2+1\GOSUB700\GOSUB800\GOSUB860
545 !" GROUNDS INTO A DOUBLE PLAY",\GOTO580
550 !" IS RETIRED",\IFRND(0)>.5THEN580\IFO1>2THEN580
560 Q=1\N=0\GOSUB700\GOSUB800\GOSUB860\GOTO582
580 !""
582 S(A,J,0)=S(A,J,0)+P\IFO1>2THEN590\GOTO210
590 !""\O1=0\O2=0\IFA=0THEN!"HOME TEAM BATTING"
592 FORR=0TO7\E(R)=0\B(R)=0\NEXT\NEXT
594 !"SCORE AFTER",G," INNINGS ",T$(1,10),R(0,0),"  ",
595 !T$(11,20),R(1,0)\!""\NEXT
596 !"          ",\FORS=1TO9\!%3I,S,\NEXT\!" TOT"
597!"========== == == == == == == == == ==  =="
598 FORS=0TO1\A=1+(10*S)\!T$(A,A+9),\FORA=1TO9\!%3I,R(S,A),\NEXT\!%4I,R(S,0
599 NEXT\IFR(1,0)<>R(0,0)THEN600\T=1\GOTO200
600 !""\INPUT"READY FOR BOX SCORE? ",Z$\!""\S(W1,W2,6)=9\S(W3,W4,6)=-9
605 !"BOXSCORE",TAB(10),T$(1,10),TAB(41),T$(11,20)\!""
610 !"NAME       POS AB HITS HR RBI   ",
615 !"NAME       POS AB HITS HR RBI"
620 FORF=0TO17\T=0\FORA=0TO1\IFW(A,F)<0THEN635\IFA=1THEN!TAB(32),
625 M=W(A,F)-1\GOSUB850\G=INT(F/2)\IFG*2<>FTHEN628
626 B=A(A,2)\G=L(A,B,G,1)*2\IFG=0THENG=2\!"  ",P$(G-1,G),\GOTO629
628 !"    ",
629 !%3I,G(A,M,0),%5I,G(A,M,1),%3I,G(A,M,2),%4I,G(A,M,3),
631 FORS=0TO3\G(A,18,S)=G(A,18,S)+G(A,M,S)\NEXT\T=1
635 NEXT\IFT>0THEN!""\NEXT
636 !"                ==  ==  =  ==                ==",
637 !"  ==  =  ==\!"TOTAL",%12I,G(0,18,0),%5I,G(0,18,1),%3I,G(0,18,2),
638 !%4I,G(0,18,3),"   TOTAL",%12I,G(1,18,0),%5I,G(1,18,1),%3I,G(1,18,2),
640 !%4I,G(1,18,3)\!" "\INPUT"READY FOR PITCHING STATS? ",Z$\!"PITCHING"
642 !"NAME        IP  H  R ER  W  K DECISION"
643 !"========== ===== == == == == == ========"\FORC=1TOOSTEP-1\FORB=0TO11
644 IFS(C,B,0)+S(C,B,1)+S(C,B,4)=0THEN649
645 A=1-C\M=18+B\GOSUB850\S(C,B,0)=S(C,B,0)/3
646 !%5F1,S(C,B,0),%3I,S(C,B,1),S(C,B,2),S(C,B,3),S(C,B,4),S(C,B,5),
647 IFS(C,B,6)>0THEN!" LOSSER",\IFS(C,B,6)<0THEN!" WINNER",\!""
649 NEXT\!""\NEXT\GOTO999
650 R=0\IFB(3)>0ANDB(2)>0ANDB(1)>0THEN700
660 IFB(2)>0ANDB(1)>0THEN700
670 IFB(1)>0ANDB(2)=0ANDB(3)=0THEN700
680 IFB(1)>0ANDB(2)=0THEN690\B(1)=J+1\RETURN
690 B(2)=B(1)\B(1)=J+1\RETURN
700 FORS=3TO1STEP-1\B(S+Q)=B(S)\B(S)=0\NEXT\B(N)=J+1\R=0
710 FORS=4TO7\IFB(S)=0THEN727\B(S)=0\IFA>9THEN725
720 R(A,G)=R(A,G)+1\R(A,0)=R(A,0)+1
725 G(A,M,3)=G(A,M,3)+1\R=R+1\S(A,J,2)=S(A,J,2)+1
727 NEXT\IFR(0,0)=R(1,0)THENRETURN
730 IFS1>S2ANDR(0,0)>R(1,0)THENRETURN
740 IFS2>S1ANDR(1,0)>R(0,0)THENRETURN
750 S=1\IFA=STHENS=0\W1=A\W2=A(A,1)\W3=S\W4=A(S,1)
790 S1=R(0,0)\S2=R(1,0)\RETURN
800 IFO2>2THENRETURN
805 FORS=3TO1STEP-1\E(S+Q)=E(S)\E(S)=0\NEXT\E(N)=J+1
810 FORS=4TO7\IFE(S)=0THEN830\E(S)=0
820 S(A,J,3)=S(A,J,3)+1
830 NEXT\RETURN
850 S=(A*300)+(M*10)+1\!N$(S,S+9),\RETURN
860 IFO1>2THENRETURN\N=0\!TAB(40),\FORS=1TO3\IFB(S)=0THEN880
865 IFN>0THEN870\!"RUNNER ON ",\N=1
870 N=S%6\!B$(N-5,N)," ",
880 NEXT\!""\RETURN
900 IFE(3)>0ANDE(2)>0ANDE(1)>0THEN800
910 IFE(2)>0ANDE(1)>0THEN800
920IFE(1)>0ANDE(2)=0ANDE(3)=0THEN800
930 IFE(1)>0ANDE(2)=0THEN690\E(1)=J+1\RETURN
940 E(2)=E(1)\E(1)=J+1\RETURN
950 INPUT"PITCHER, HITTER OR BOTH? ",Z$\IFZ$=""THENRETURN
952 IFZ$="H"THEN960
954 INPUT"ID# OF NEW PITCHER? ",S\IFS<>0THEN956\!"NO CHANGE"\GOTO960
956 IFS<10RS>12THEN954\J=S-1\A(A,1)=J
958 FORF=0TO4\T(A,F)=P(A,S,F)\NEXT\!"RELIEF PITCHER IN"
960 IFZ$="H"THEN962\IFZ$="B"THEN962\RETURN
962 INPUT"BATTING ORDER POSITION, ID#? ",S,F\IFS<>0THEN964\!
        "NO CHANGE"\RETURN
964 IFS>=1ANDS<=9THEN966\!"INVALID BATTING POSITION"\GOTO962
966 IFF>=1ANDF<=18THEN968\!"INVALID ID#"\GOTO962
968 S=((S-1)*2)+1\IFW(A,S)=-1THEN980
970 !"YOU ALREADY MADE A SUB IN THAT POSITION"\GOTO962
980 W(A,S)=F\!"SUBSTUTITION MADE"\GOTO962
999 INPUT"RETURN TO END OR ANYTHING ELSE FOR ANOTHER GAME? ",Z$
1000 FORA=0TO1\FORB=0TO3\G(A,18,B)=0\NEXT\NEXT
1010 FORA=0TO1\FORB=0TO17\FORC=0TO3\H(A,B,C+7)=H(A,B,C+7)+G(A,B,C)
1020 G(A,B,C)=0\NEXT\NEXT\FORB=0TO11\FORC=0TO5
1030 P(A,B,C+5)=P(A,B,C+5)+S(A,B,C)\S(A,B,C)=0\NEXT
1040 IFS(A,B,6)<0THENP(A,B,11)=P(A,B,11)+1
1050 IFS(A,B,6)>0THENP(A,B,12)=P(A,B,12)+1\S(A,B,6)=0\NEXT
1060 FORB=0TO9\R(A,B)=0\NEXT\NEXT
1070 IFZ$<>""THEN110
1100 FORA=0TO1\FORB=0TO17\C=(B*72)+52
1110 WRITE#AZC,H(A,B,7),H(A,B,8),H(A,B,9),H(A,B,10),NOENDMARK\NEXT
1115 D=0\IFA=0THEND=1
1120 FORB=0TO11\C=(B*82)+1338
1130 WRITE#AZC,P(D,B,5),P(D,B,6),P(D,B,7),P(D,B,8),P(D,B,9),P(D,B,10)
1140 WRITE#A,P(D,B,11),P(D,B,12),NOENDMARK\NEXT\CLOSE#A\NEXT
1150 !FREE(0)\CHAIN"BB"
```

Program BB-Stand - Prints Standings

```
10 DIMN$(100),P$(200),L(4,5),T(10,2),T$(100)
20 FORA=10TO100STEP10\N$(A-9,A)="----------"\NEXT
25 T$=N$\P$=T$+N$
30 OPEN#0,"BBLEAG"
40 INPUT"ENTER 0 TO SET OR CHANGE LEAGUE FILE ? ",A\IFA<>0THEN300
50 INPUT"HOW MANY TEAMS IN THE LEAGUE ? ",A
55 IFA>0ANDA<11THEN60\!"MAX IS 10 TEAMS"\GOTO50
```

```
60 !'GIVE THE FILE NAME FOR THE',A,' TEAMS IN THE LEAGUE'
70 FORB=1TOA\C=B*10\!%3I,B,
75 F$='----------'
80 INPUT'  ? ',F$\N$(C-9,C)=F$\NEXT
90 WRITE#0,A,N$\GOTO310
300 READ#0,A,N$
310 CLOSE#0
315 A1=A
320 FORB=1TOA\C=10*B\F$=N$(C-9,C)\FORC=10TO1STEP-1
330 IFF$(C,C)<>'-'THENEXIT350
340 F$(C,C)=' '\NEXT
350 OPEN#0,F$
365 FORV=1TO18
370 READ#0,F$,E,E,E,E,E,E,E,E,E1,E2,E3,E4\E5=0\IFE1>0THENE5=E2/E1
380 E(1)=E5\E(2)=E3\E(3)=E4\E(4)=E2\GOSUB390\GOTO511
390 FORF=1TO4\FORG=1TO5\IFL(F,G)>=E(F)THEN500
395 I=(F-1)*50
400 FORH=4TOGSTEP-1\L(F,H+1)=L(F,H)
410 J=((H+1)*10)+I\P$(J-9,J)=P$(J-19,J-10)\NEXT
415 I=I+(10*G)\P$(I-9,I)=F$\L(F,G)=E(F)
420 EXIT 510
500 NEXT
510 NEXT\RETURN
511 NEXT\CLOSE#0\NEXT
750 !''\!'  '
800 !'LEAGUE LEADERS'
802 !'================'
810 !'NAME         AVE  NAME          HR  NAME         RBI
820 !'========== ====  ========== ====  ========== ====
          NAME        HITS'
          ========== ====='
830 FORC=1TO5\FORA=1TO4\B=(50*(A-1))+(C*10)
840 !P$(B-9,B),\IFA>1THEN860
850 !%5F3,L(A,C),'  ',\GOTO870
860 !%5I,L(A,C),'  ',
870 NEXT\!' '\NEXT
905 FORV=1TO4\FORE=1TO5\L(V,E)=-9999\NEXT\NEXT
910FORB=1TOA1\C=10*B\F$=N$(C-9,C)\FORC=10TO1STEP-1
920 IFF$(C,C)<>'-'THENEXIT950
930 IFF$(C,C)<>'-'THENEXIT950
940 F$(C,C)=' '\NEXT
950 OPEN#0,F$
951 C=1291\READ#0%C,E
952 FORV=1TO12\READ#0,F$,E,E,E,E,E,E,E1,E2,E3,E4,E5,E6,E7,E8
954 E(4)=0\IFE1>0THENE(4)=(9*E4)/E1\E(2)=E6\E(3)=E7\E(1)=0
955 IFE7>0THENE(1)=E7/(E7+E8)\E(4)=0-E(4)
956 T(B,0)=T(B,0)+E7\T(B,1)=T(B,1)+E8
960 IFE1=0THENE(4)=-99999
980 GOSUB390
990 NEXT\CLOSE#0\NEXT
995 !''\!''
1000!'LEAGUE LEADERS'
1002!'================'
1004!'NAME         PCT  NAME          KO  NAME         WINS
1006!'========== ====  ========== ====  ========== ====
          NAME          ERA'
          ========== ======='
1008FORC=1TO5\FORA=1TO4\B=(50*(A-1))+(C*10)
1010!P$(B-9,B),\IFA>1ANDA<4THEN1014\IFA=4THEN1015
1012!%5F3,L(A,C),'  ',\GOTO1016
1014!%5I,L(A,C),'  ', \GOTO1016
1015 !%7F2,0-L(A,C),
1016NEXT\!' '\NEXT
1200 FORA=1TOA1\S(A)=A\T(A,2)=T(A,0)/(.00001+T(A,0)+T(A,1))\NEXT
1210 B=0\FORA=1TOA1-1\C=S(A)\C1=S(A+1)
1220 IFT(C,2)>=T(C1,2)THEN1240
1230 S(A+1)=C\S(A)=C1\B=10
1240 NEXT\IFB>0THEN1210
1250 !''\INPUT'READY FOR STANDINGS ? ',Z$\!''\!'STANDINGS'\!'=========='
1260 !'TEAM        WINS LOST   PCT    GB'
1270 !'========== ==== ==== ===== ====='
1280 C=S(1)\C=T(C,0)-T(C,1)
1290 FORA=1TOA1\E=S(A)\F=10*E\G=C-(T(E,0)-T(E,1))\G=G*.5
1300 !N$(F-9,F),%5I,T(E,0),T(E,1),%6F3,T(E,2),%6F1,G\NEXT
1305 !''\!''
1310 !FREE(0)\CHAIN'BB'
```

Program BB - Links Programs

```
10 !'THIS IS THE BASEBALL SIMULATION.  THE COMMANDS ARE:'
   !'       0  TO END'
   !'       1  TO PLAY A GAME'
30 !'       2  TO DISPLAY TEAM STATISTICS'
40 !'       3  LEAGUE STANDINGS DISPLAY'
50 !'       4  LIST TEAM ROSTER'
60 !'       5  INPUT OR CORRECT A ROSTER'
70 !'       6  COMPLETELY CLEAR A FILE'
80 !'       7  CLEAR STATISTICS ONLY FROM A FILE'
90 INPUT'       COMMAND NUMBER ? ',A\IFA>0THEN110
100 !FREE(0)\END
110 IFA>7THEN90
120 ONAGOTO131,132,133,134,135,136,137
131 CHAIN'BB-GAME'
132 CHAIN'BB-STAT'
133 CHAIN'BB-STAND'
134 CHAIN'BB-ROST'
135 CHAIN'BB-INPUT'
136 CHAIN'BB-S-F'
137 CHAIN'BB-C-F'
```

Program BB-s-F
Initializes New Team Files

```
5 DIMN$(10)
10 INPUT'FILE TO BE CLEARED: ',N$
20 !'IF YOU ENTER A RETURN FILE ',N$,' WILL BE COMPLETELY ERASED ',
30 INPUTZ$\IFZ$<>''THENCHAIN'BB'
40 OPEN#0,N$\N$='.........'
50 FORB=1TO18\WRITE#0,N$,A,A,A,A,A,A,A,A,A,A,A,A\NEXT
60 FORB=1TO12\WRITE#0,N$,A,A,A,A,A,A,A,A,A,A,A,A,A\NEXT
70 FORB=1TO54\WRITE#0,A\NEXT
80 !FREE(0)\CHAIN'BB'
```

Program BB-C-F -
Clears Results of Previous Simulations

```
5 DIMN$(10)
10 INPUT'FILE TO BE CLEARED: ',N$
20 !'IF YOU ENTER A RETURN FILE ',N$,' WILL HAVE ITS
        STATISTICS REMOVED',
30 INPUTZ$\IFZ$<>''THENCHAIN'BB'
40 OPEN#0,N$\N$='.........'
50 FORB=0TO17\C=(72*B)+52\WRITE#0%C,A,A,A,A,NOENDMARK\NEXT
60 FORB=0TO11\C=(82*B)+1338\WRITE#0%C,A,A,A,A,A,A,A,A,NOENDMARK\NEXT
80 !FREE(0)\CHAIN'BB'
```

Program BB-STAT -
Prints Statistics

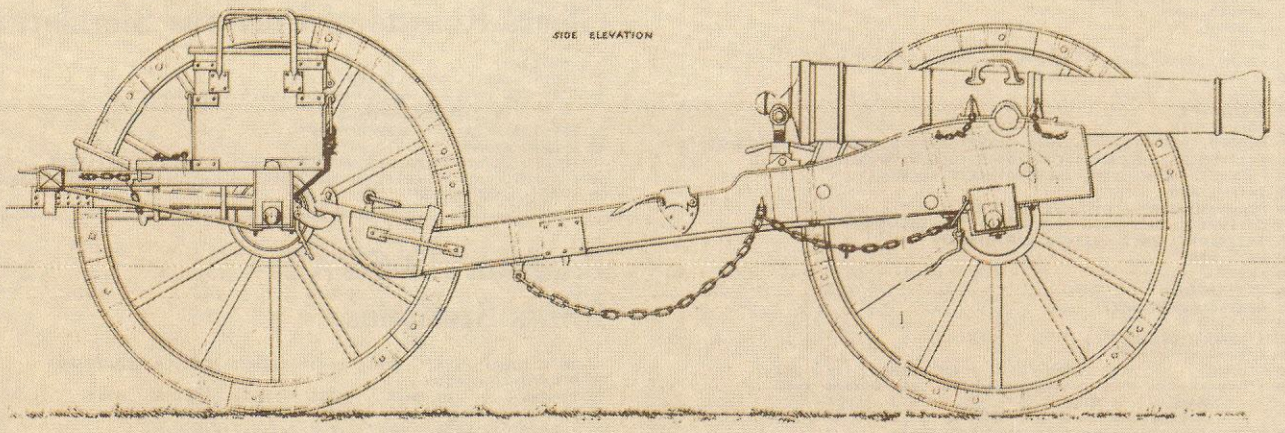
```
10 INPUT'WHAT FILE? ',Z$\IFZ$<>''THEN20\!FREE(0)\CHAIN'BB'
20 OPEN#0,Z$\LINE80
30 !'NAME        AT BATS    HITS    H.R.   RBI'S   AVE'
40 !'========== ======= ====== ====== ====== ====='
50 FORA=0TO17\READ#0,N$,Z,Z,Z,Z,Z,Z,Z,Z,X,C,V,B
60 Q=Q+X\W=W+C\E=E+V\R=R+B
70 !N$,%8I,X,C,V,B,\V=0\IFX>0THENV=C/X\!%6F3,V\NEXT
75 !'========== ======= ====== ====== ====== ====='
80 N$='TOTALS....'\!N$,%8I,Q,W,E,R,%6F3,W/(Q+.0000001)
90 Q=0\W=0\E=0\R=0
92 !''\INPUT'READY FOR PITCHERS? ',Z$\!''
94  !'PITCHERS      I.P.   HITS   RUNS   E.R.  WALKS   KO'S  WON LOST',
96 !'   PCT    ERA'
100 !'========== ======= ====== ====== ====== ====== ====== ==== ====',
105 !' ===== ======='
110 FORA=0TO11\READ#0,N$,Z,Z,Z,Z,Z,Z,X,C,V,B,N,M,K,L
120 Q=Q+X\W=W+C\E=E+V\R=R+B\T=T+N\Y=Y+M\U=U+K\I=I+L
122 O=0\P=K+L\IFF>0THENO=K/P
124 P=0\IFX>0THENP=(9*B)/X\IFP>99.99THENP=99.99
130 !N$,%8F1,X,%7I,C,V,B,N,M,%5I,K,L,%6F3,O,%7F2,P
140 NEXT
150 !'========== ======= ====== ====== ====== ====== ====== ==== ====',
155 !' ===== ======='
160 O=U/(U+I+.000001)\P=(9*R)/(Q+.00001)\N$='TOTALS....'
170 !N$,%8F1,Q,%7I,W,E,R,T,Y,%5I,U,I,%6F3,O,%7F2,P\!''
180 Q=0\W=0\E=0\R=0\T=0\Y=0\U=0\I=0\CLOSE#0\GOTO10
```

Program Input

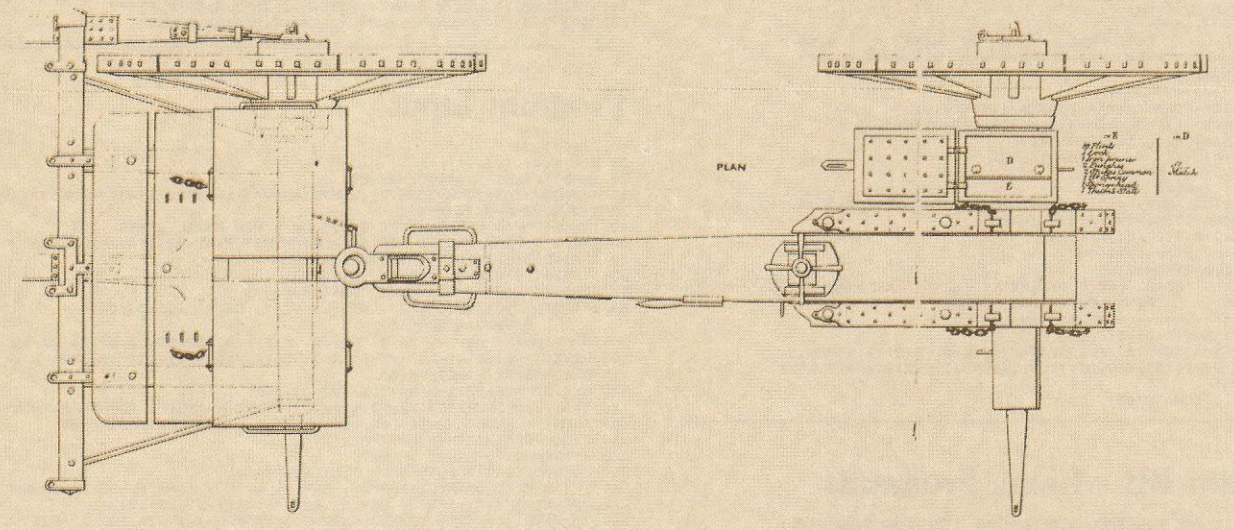
```
10 DIMO(1,9)
15 B$='.........'
20 INPUT'FILE: ',N$\OPEN#0,N$
30 A=38037/143974\W=13757/(13757+143974)\K=21722/(143974-38037)
35 I=2.8071\H=3644/38037
40 INPUT'O FOR WEEKLY TYPE STATS OR 1 FOR FINAL STATS? ',T
50 INPUT'# FROM AND TO FOR NON-PITCHERS? ',A1,A2\IFA1>A2THEN50
60 IFA1=0ORA1>18THEN200\IFA2>18THEN50
65 !'BATS: 0-RIGHT  1-LEFT  2-BOTH'
70 FORC=A1TOA2\INPUT'NAME? ',N$\N$=N$+B$
80 IFT>0THEN90\INPUT'BATS, AB, H, D, T, HR ? ',D1,D2,D3,
        D4,D5,D6\GOTO100
90 INPUT'BATS, AB, H, D, T, HR, W, K ? ',D1,D2,D3,D4,D5,D6,D7,D8
100 R1=(D3/D2)-A\R2=D4/(D3-D6)\R3=R2+(D5/(D3-D6))\R4=(D6/D3)-H
110 R5=0\R6=0\IFT=0THEN130
120 R5=(D7/(D7+D2))-W\R6=(D8/(D2-D3))-K
130 Z=(C-1)*72\WRITE#0%Z,N$,D1,Z,R1,R2,R3,R4,R5,R6,NOENDMARK\NEXT
200 INPUT'# FROM AND TO FOR PITCHERS? ',A1,A2\IFA1>A2THEN50
210 IFA1=0ORA1>12THEN300\IFA2>12THEN60
220 !'THROWS: 0-RIGHT 1-LEFT'
230 FORC=A1TOA2\INPUT'NAME? ',N$\N$=N$+B$
240 IFT>0THEN250\INPUT'THROWS, IP, H, W, K ? ',D6,D1,D2,D3,D4\GOTO260
250 INPUT'THROWS, IP, H, HR, W, K ? ',D6,D1,D2,D5,D3,D4
260 R5=(I*D1)+D2\R1=(D2/R5)-A\R2=(D3/(D3+R5))-W\R3=(D4/(R5-D2))-K
270 R4=(D5/D2)-H\IFT=0THENR4=0
275 Z=((C-1)*82)+1296\WRITE#0%Z,N$,D6,Z,R1,R2,R3,R4,NOENDMARK\NEXT
300 !''\!'INPUT LINE-UPS'
310 INPUT'THERE ARE 3 LINE-UPS, WHICH ONE IS TO BE UPDATED
        OR 0 TO END ? ',L
320 IFL=0THEN400\IFL>3THEN310\L=L-1
330 !'INPUT PLAYER ID#, POS #'\FORA1=1TO9\GOTO340
335 !'INVALID ENTRY, REDO BATTING POSITION',A1
340 INPUTR1,R2\IFR1<0ORR1>18THEN335\IFR2<0ORR2=10RR2>9THEN335
350 O(0,A1)=R1\O(1,A1)=R2\NEXT
360 Z=2280+(L*90)\WRITE#0%Z,O(0,1),O(1,1)
370 FORA1=2TO9\WRITE#0,O(0,A1),O(1,A1),NOENDMARK\NEXT\GOTO300
400 CLOSE#0\!FREE(0)\CHAIN'BB'
```

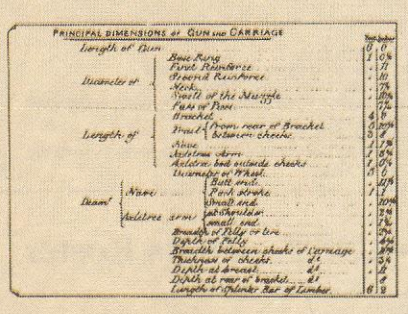
Program BB-ROST - Prints Roster

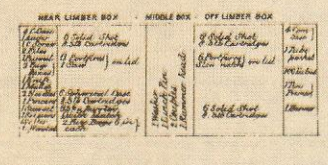
```
10 DIMN$(10),N(2,8),P$(18),P(2,8),O$(180)
15 P$="DH C1B2B3BSSLFCFRF"
20 INPUT'FILE: ',N$\OPEN#0,N$\!''
30 !' # NAME        B     H     D     T    HR     W     K'
40 !'== ========== = ===== ===== ===== ===== ===== ===== ====='
50 FORA=1TO18\READ#0,N$,Z,S,X,C,V,B,N,M,S,S,S,S\S=A*10\O$(S-9,S)=N$
60 !%2I,A,' ',N$,%2I,Z,%6F3,X,C,V,B,N,M\NEXT\!''\INPUT'READY? ',Z$\!''
70 !' # NAME       T     H     W     K    HR'
80 !'== ========== = ===== ===== ===== ====='
90 FORA=1TO12\READ#0,N$,Z,S,X,C,V,B,S,S,S,S,S,S,S,S
100 !%2I,A,' ',N$,%2I,Z,%6F3,X,C,V,B\NEXT\!''\INPUT'READY? ',Z$\!''
110 FORA=0TO2\FORB=0TO8\READ#0,P(A,B),N(A,B)\NEXT\NEXT
120 !'LINE-UP NUMBER 1   LINE-UP NUMBER 2   LINE-UP NUMBER 3'
130 !'================   ================   ================'
140 FORA=0TO8\FORB=0TO2\C=2*N(B,A)\IFC<2THENC=2\S=P(B,A)*10
150 !O$(S-9,S),'  ',P$(C-1,C),'    ',\NEXT\!''\NEXT
160 CLOSE#0\!''\!FREE(0)\CHAIN'BB'
```



GUNNER GAME

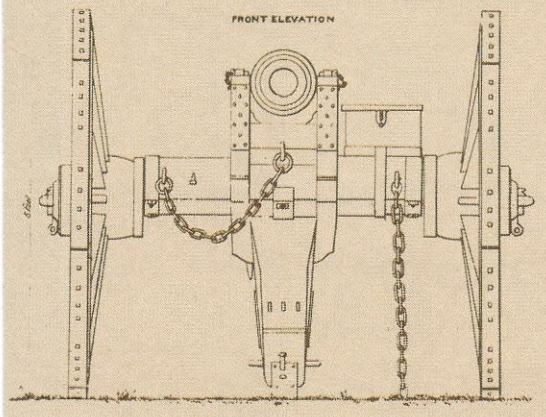






PROGRAM LISTING

```
10 REM "GUNNER" WRITTEN IN NORTHSTAR BASIC V6 R3.
20 INPUT "DO YOU NEED INSTRUCTIONS (YES OR NO)? ",C$
30 IF C$(1,1)="N" THEN 130
40 PRINT "     YOU ARE THE COMMANDER OF A LONG RANGE (10KM)"
50 PRINT "CANNON.  A RANDOM NUMBER OF TARGETS WILL BE REPORTED"
60 PRINT "BY A FOWARD OBSERVER.  THE RANGE OF EACH TARGET WILL "
70 PRINT "BE GIVEN.  YOUR MISSION IS TO DESTROY EACH TARGET WITH"
80 PRINT "AS FEW ROUNDS POSSIBLE.  A HIT WITHIN 25 METERS OF"
90 PRINT "THE TARGET WILL DESTROY IT.  THE ENEMY"
100 PRINT "TARGET WILL BE RETURNING FIRE, SO BE CAREFUL!"
110 PRINT "YOUR EFFICIENCY WILL BE SCORED AT THE END OF THE"
120 PRINT "MISSION, IF YOU LIVE THAT LONG."
130 PRINT "ENTER ELEVATION IN DEGREES AS INSTRUCTED! GOOD LUCK!"
140 R=0\REM NUMBER OF ROUNDS EXPENDED.
150 PRINT
160 INPUT "ENTER YOUR RATING!  (1(EXPERT)-10(NOVICE)) :",K1
170 PRINT
180 INPUT "ENTER A RANDOM NUMBER :",K\REM NO RANDOMIZE AVAIL.
190 K=ABS(K)
200 IF K<1 THEN 230
```



BY LAURENCE G. DISHMAN

Few artillery programs provide automatic return fire routines for two reasons. First, such routines require arcsine functions to calculate return firing angles and few BASIC interpreters have this feature. Second, if the arcsine were provided, the machine could "guess" the correct angle first time, every time.

Gunner solves both these problems effectively by using the Maclaurin series for the arcsine function in the subroutine at Steps 970-1090. This subroutine calculates any arcsine in the fourth and first quadrant in about one second. To this angle, a random error is added, based on the rating of the operator. A rating of 1 will provide a rather small error while a 10 will generate a substantial first shot error.

Steps 880-960 provide an error correction routine that on occasion makes mistakes itself (a most human failing).

All equations used describe theoretical cases, neglecting air resistance, wind, terrain inhomogeneities and so forth. A range of 10 kilometers requires a shell velocity of 315 meters per second.

The timer routine in Steps 790-800 causes a delay analogous to the time necessary for shell travel. It's interesting to watch an operator "sweat out" incoming enemy rounds while wondering if he's going to lose the game or get another shot at victory.

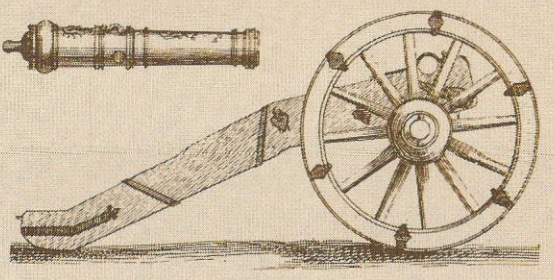
Finally, a score of 25 percent may not seem worthy of congratulations; however, one hit in four tries isn't a bad average – when someone's shooting back at you. Try it! □

Listing continued

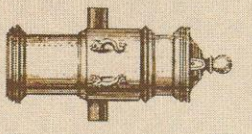
```
210 K=K/10
220 GOTO 200
230 A=RND(K)\REM K USED AS RND SEED ONLY IF 0<K<1.
240 PRINT
250 K=0\REM K NOW USED TO COUNT DESTROYED TARGETS.
260 N=INT(RND(0)*8)+3\REM NUMBER OF TARGETS 3<=N<=10.
270 FOR N1=1 TO N
280 Q=0\REM Q USED IN RETURN FIRE SUBROUTINE.
290 D=INT(RND(0)*5000)+5000\REM DISTANCE TO TARGET 5KM<D<10KM.
300 PRINT "DISTANCE TO TARGET: ";%05I,D," METERS."
310 INPUT "ENTER ELEVATION: ",A1
320 IF A1>1 AND A1<89 THEN 330
330 PRINT
340 PRINT "ELEVATION MUST BE BETWEEN 1 AND 89 DEGREES."
350 PRINT "ENEMY POSITION GETS A FREE SHOT AT YOU!"
360 PRINT
370 GOTO 460
380 A=A1/57.296\REM CONVERT TO RADIANS.
390 R=R+1
400 GOSUB 740\REM TO CALC DIST, TIME, ALT AND ERROR.
410 IF ABS(E)<25 THEN 500\REM IF ERROR<25 M, TARGET DESTROYED.
420 PRINT
430 X1=X\T4=T\H1=H\E1=E\REM TRANSFER SUBR VAR TO DISPLAY VARS.
440 PRINT "TARGET RETURNING FIRE!"
450 PRINT
460 GOSUB 820\REM GET RETURN FIRE ANGLE.
470 A=A2/57.296
480 GOSUB 740
490 X2=X\REM SAVE ENEMY SHELL DISTANCE.
500 E=ABS(E)\IF E<25 THEN 700\REM IF E<25 M, TOO BAD!
510 PRINT "MISSED YOU BY: ";%05I,E," METERS."
520 PRINT "TIME OF FLIGHT: ";%5F2,T4," SECONDS."
530 PRINT "MAXIMUM ALTITUDE: ";%05I,H1," METERS."
540 PRINT "DISTANCE TRAVELLED: ";%06I,X1," METERS."
550 PRINT "LAST ELEVATION: ";%5F2,A1," DEGREES."
560 PRINT "CORRECT: ";%07I,E1," METERS."
570 GOTO 800
580 PRINT "  *** TARGET DESTROYED ***"
590 PRINT
600 PRINT
610 K=K+1
620 NEXT N1\REM GET NEXT TARGET.
630 PRINT "NO MORE ENEMY POSITIONS REPORTED."
640 PRINT "YOU DESTROYED: ";%2I,K," ENEMY POSITIONS WITH"
650 PRINT %3I,R," ROUNDS EXPENDED."
660 P=(K/R)*100
670 PRINT "SCORE: ";%6F2,P,"%."
680 IF P>= 25. THEN PRINT "GOOD SHOOTING!"
690 END
700 PRINT "YOUR POSITION HAS BEEN DESTROYED.  "
710 FOR I=1 TO 800\NEXT I
720 PRINT "TOO BAD!"
730 END
740 X=INT((99225*SIN(2*A))/9.8)\REM DISTANCE SHELL TRAV.
750 T=(630*SIN(A))/9.8\REM TIME OF SHELL TRAVEL.
760 H=INT((99225*(SIN(A)^2))/19.6)\REM MAX ALTITUDE OF SHELL.
770 E=X-D\REM ERROR IN METERS.
780 T1=INT(T)
790 FOR T2=1 TO T1 STEP 0.01
800 NEXT T2\REM WAIT FOR SHELL TO HIT.
810 RETURN
820 IF Q=0 THEN 830 ELSE ON Q GOTO 880,920,950
830 A3=D*9.877E-5
840 A3=FNA(A3)*28.64789
850 A2=A3-(RND(0)*K)\REM 1ST ANGLE APPROX. THE LARGER K)
860 Q=1                 \REM THE LARGER THE ANGLE ERROR.
870 RETURN
880 A4=A2+((A3-A2)/(RND(0)+1))
890 IF A4>(A2+0.1) THEN A2=A4 ELSE GOTO 910
900 RETURN
910 Q=2
920 IF D>X2 THEN A2=A2+0.5 ELSE GOTO 940
930 RETURN
940 Q=3
950 IF D<X2 THEN A2=A2-0.1 ELSE A2=A2+0.05
960 RETURN
970 DEF FNA(X)\REM THIS FUNCTION RETURNS THE ARCSINE IN RADIANS
980 IF X>0.5 THEN 1060
990 A=X\D1=1\N=1
1000 FOR I=2 TO 10
1010 D1=(2*I-2)*D1
1020 N=(2*I-3)*N
1030 A=A+((N*X^(2*I-1))/(D1*(2*I-1)))
1040 NEXT I
1050 RETURN A
1060 Y=SQRT((1-X)*0.5)
1070 A=(-4*FNA(Y)+3.1415927)*0.5
1080 RETURN A
1090 FNEND
```

(Sample Run on following page)

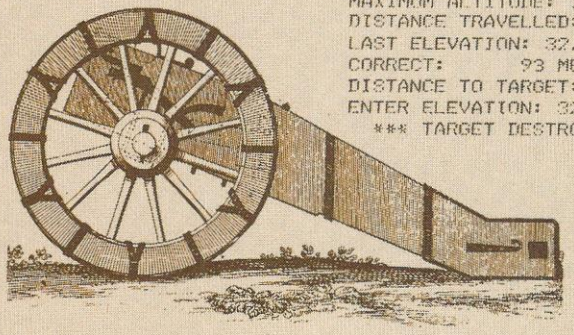
SAMPLE RUN



```
DO YOU NEED INSTRUCTIONS (YES OR NO)? YES
     YOU ARE THE COMMANDER OF A LONG RANGE (10KM)
CANNON.  A RANDOM NUMBER OF TARGETS WILL BE REPORTED
BY A FOWARD OBSERVER.  THE RANGE OF EACH TARGET WILL
BE GIVEN.  YOUR MISSION IS TO DESTROY EACH TARGET WITH
AS FEW ROUNDS POSSIBLE.  A HIT WITHIN 25 METERS OF
THE TARGET WILL DESTROY IT.  THE ENEMY
TARGET WILL BE RETURNING FIRE, SO BE CAREFUL!
YOUR EFFICIENCY WILL BE SCORED AT THE END OF THE
MISSION, IF YOU LIVE THAT LONG.
ENTER ELEVATION IN DEGREES AS INSTRUCTED! GOOD LUCK!

ENTER YOUR RATING!  (1(EXPERT)-10(NOVICE)) :7

ENTER A RANDOM NUMBER :789

DISTANCE TO TARGET: 5,035 METERS.
ENTER ELEVATION: 15

TARGET RETURNING FIRE!

MISSED YOU BY: 1,971 METERS.
TIME OF FLIGHT: 16.64 SECONDS.
MAXIMUM ALTITUDE:   339 METERS.
DISTANCE TRAVELLED:  5,062 METERS.
LAST ELEVATION: 15.00 DEGREES.
CORRECT:      27 METERS.
DISTANCE TO TARGET: 5,035 METERS.
ENTER ELEVATION: 14.9
  *** TARGET DESTROYED ***
```



```
DISTANCE TO TARGET: 9,083 METERS.
ENTER ELEVATION: 32.5

TARGET RETURNING FIRE!

MISSED YOU BY: 1,102 METERS.
TIME OF FLIGHT: 34.54 SECONDS.
MAXIMUM ALTITUDE: 1,461 METERS.
DISTANCE TRAVELLED:  9,176 METERS.
LAST ELEVATION: 32.50 DEGREES.
CORRECT:      93 METERS.
DISTANCE TO TARGET: 9,083 METERS.
ENTER ELEVATION: 32.0
  *** TARGET DESTROYED ***
```



```
DISTANCE TO TARGET: 5,404 METERS.
ENTER ELEVATION: 15.4

TARGET RETURNING FIRE!

MISSED YOU BY: 1,815 METERS.
TIME OF FLIGHT: 17.07 SECONDS.
MAXIMUM ALTITUDE:   357 METERS.
DISTANCE TRAVELLED:  5,134 METERS.
LAST ELEVATION: 15.40 DEGREES.
CORRECT:     -220 METERS.
DISTANCE TO TARGET: 5,404 METERS.
ENTER ELEVATION: 15.7

TARGET RETURNING FIRE!

MISSED YOU BY:   883 METERS.
TIME OF FLIGHT: 17.40 SECONDS.
MAXIMUM ALTITUDE:   370 METERS.
DISTANCE TRAVELLED:  5,275 METERS.
LAST ELEVATION: 15.70 DEGREES.
CORRECT:     -129 METERS.
DISTANCE TO TARGET: 5,404 METERS.
ENTER ELEVATION: 16.3

TARGET RETURNING FIRE!

MISSED YOU BY:   275 METERS.
TIME OF FLIGHT: 18.04 SECONDS.
MAXIMUM ALTITUDE:   398 METERS.
DISTANCE TRAVELLED:  5,455 METERS.
LAST ELEVATION: 16.30 DEGREES.
CORRECT:      51 METERS.
DISTANCE TO TARGET: 5,404 METERS.
ENTER ELEVATION: 16.2
  *** TARGET DESTROYED ***

NO MORE ENEMY POSITIONS REPORTED.
YOU DESTROYED:  3 ENEMY POSITIONS WITH
  8 ROUNDS EXPENDED.
SCORE:  37.50%.
GOOD SHOOTING!
```

Terminate Your System... For Life

BY CHIP A. TYETI

Whether your computer is in a terminal dilemma, suffers from a sick baud rate or even has "interfacial" complexion problems, you'll always need to communicate with your micro via an efficient terminal. While some terminals are built into microcomputers, most are physically separate units, often interfacing via RS232 or 20 mA current loop. There's a large selection of terminals as well as CRTs and keyboards. We'll review some of the characteristics of all three peripheral creatures, then discuss features of many of the units presently on the market.

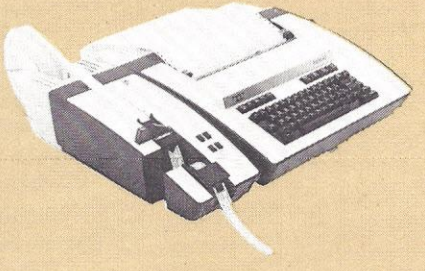
A terminal may contain a CRT display or a hard-copy printer. Microprocessor-controlled terminals are called intelligent terminals; a dumb terminal is a slave to your system. For communications over telephone lines a modem (modulator-demodulator) is essential and may already be built into the terminal. Occasionally an acoustic coupler linking the modem to the telephone's handset is also built-in, usually at the back of the unit. Portable terminals, in cases resembling suitcases, often feature acoustic couplers on one side of the printer mechanism.

Text editing on a terminal may be done either by line or by block. A floppy disk system incorporated into the terminal can use one, two, or more drives for 8″ or 5-1/4″ disks or a combination of both sizes.

CRTs may scroll up, down, left, right, or in any combination of moves. Some VDTs (Video Display Terminals) scroll one line at a time. This scrolling appears as a jerky, jumping motion and high scrolling rates may prove annoying. Many VDTs use a scan scroll which causes the line of characters to move off the screen in fractional degrees, gradually and smoothly.

Graphic display resolution refers to the number of lines and number of characters per line that will fit onto the CRT. Locating data in memory relative to the video display can use one of three mapping techniques – character ROM, bit map or vector mapping. The baud rate – the rate of data transmission to the terminal – is one of the most significant factors in selecting your terminal. If a terminal is not com-

OSI's 540
Video Display
with Keyboard



The Teletype 8-Level
Basic Model 43
ASR Teleprinter

Computer Transceiver Systems'
Execuport 4000

Micro-Terms ACT
Terminals

TEL's MCS-PT112
From CMC Marketing

George Risk Industries
771 Keyboard

patible with the baud rates of your micro, you'll find interfacing difficult.

Some terminals incorporate special features such as inverse video, character blink, half intensity, protect area, and a variety of display colors beyond the standard black and white or black and green.

In inverse video, the field is reversed. Where white was, black is now; where black was, white is now. Although we were weaned on black-on-white images, many terminals use white-on-black. Actually, the white (or green) on black is often sharper and easier to read.

Character blink allows one or more characters to blink in unison with the main cursor, or out of synchronization with the main cursor. Some terminals permit entire words, paragraphs or columns to blink. Such a CRT display can provide effective output – if not used excessively.

Half intensity is just what it sounds like: specific characters displayed at half the intensity or brightness of the standard character. A protect area prohibits writing within its boundaries and is often combined with the half intensity feature to provide a repeated format (e.g., invoice, purchase order, inventory listing, payroll records, etc.).

Only a few terminals offer color versatility. Compucolor recently introduced a CRT, keyboard and floppy disk drive integrated into one simple, attractive unit.

Half duplex and full duplex refer to data transmission. If the data can travel in only one direction at a time, the system is half duplex. If data can bounce between the two units simultaneously, the terminal has full duplex capability.

The CRT used in the various terminals may be either a monitor, a standard TV (which must be modified or equipped with an RF modulator) or an internally modified TV. Internally modified TVs use the direct video input method of data transmission vs. the RF transmitter method, where leads are simply attached to the antenna terminal posts on the back of the television.

As you can see from the chart, screen size varies from 5″ to 15″ (and even to a 7′ diagonal screen used in Advent's Videobeam projection TV). For displays smaller than 5″ diagonal, you need acute vision. I had no trouble, though, in seeing the smaller credit lines scrolling up at the end of a television program (standard TV) on the Sinclair 2″ screen, but 80 lines of 24 characters each may be just too much (or rather, too little).

Using maximum character display is not very practical if you can't see the words or characters – no matter how large the screen may be. Maybe forty lines or so of about 12 or 15 characters each will prove easier to read. Remember, the CRT doesn't necessarily carry the same configuration for the data that a hard-copy printer would. If your printer has 80 columns and your CRT display uses 40, then probably less than half the data your printer will run per page will appear on the screen of your CRT.

In preparing our chart, we asked whether the CRT units (either alone or as part of terminals) were hot chassis or isolated. Hot chassis (the chassis is the ground for one of the "hot" lines from the AC cord) have all but disappeared from the current TV marketplace. A few years back a large majority of the small transistor TVs were hot chassis. You can wire these TVs to eliminate the hot feature, but alternatively you should consider just keeping the enclosure on the set and using an RF modulator to input the video signal.

If the CRT is not a TV or never was, you may be frustrated sometime in the future for not being able to pump an audio signal out to the CRT. You may only be concerned with a tone generator or just an audio jack. Be aware of these possibilities, especially when contemplating the purchase of a "silent" CRT.

Your terminal's character matrix may not be too important to you as long as you can distinguish the characters on the screen, but check further. Does the display show both upper and lower case? Do the descenders (letters going below the line, such as "g") actually descend or just sit on the same horizontal as the non-descending characters? What about ascenders (letters going above the line, such as "d")?

Three additional CRT specifications affect adaptation of a TV to use as a CRT: the peak-to-peak composite video input voltage, the bandwidth and the impedence. As the chart shows, few companies supplied these specs because they're not important in complete, ready-to-use systems. If you plan to modify a standard TV, check your local computer store for either RF modulators (like the ones available from ATV Research) or direct video input interfaces (like the one available from Pickles and Trout).

Generally, the power requirements noted in the CRT charts refer to standard house current or available current from your system.

Keyboards are usually included in terminals – although the keyboard may be just a little 4 x 4 hex pad. Whether you buy a keyboard separately or have one built into your terminal, you should review its configuration. We've noticed a predominance of either ASCII or TTY keyboards. There are scores of others, but generally you will see the 8-level (or 8-bit) ASCII or the 5-level TTY code configured on the keyboard in various patterns. In most ASCII and TTY keyboards, the keys are placed in specific locations; but again, there is some variation in the field.

Computer Data Systems' Versatile 3B with Micropolis Drive

Telpar's PS-48 C Interactive I/O Terminal/Printer

Southwest Technical Products' CT-64 Terminal

Check the layout of the surplus keyboards before you conclude that a specific configuration is unavailable. You can rewire some boards to include missing characters, but don't bet on it.

Your computer's main power supply – with or without slight modification – will probably supply your keyboard's voltage requirements.

Whether you have a negative or positive data strobe depends on how and when your micro will accept data. The number of data lines will vary, but as long as there are enough lines for adequate parallel or serial data being transmitted, the additional lines are primarily optional as assists to the data transmission.

Of course, you can purchase keyboards either in kit form or fully assembled and ready to interface with your terminal or micro.

Keyboard mechanisms vary from Hall effect types to capacitive key switches to mechanical gold crosspoint keys. Some users seem to prefer one type over another. I have no personal preference, as long as the action is fast, clean and relatively error-free.

Now, before you glean every bit of data from the accompanying charts, I want you to know how they are related to one another. First, Charts 1 to 4 include terminals only. Charts 5 and 6 refer to CRTs; and Charts 7 and 8 to keyboards. As you can see, essential data on CRTs and on keyboards have been incorporated into the charts on terminals. Within the CRT charts is a special listing of devices used for video input, either through direct video or RF modulation.

We've included in this article terminals that are only printed circuit boards. Once you interface them to a CRT, printer and/or keyboard, you'll have a complete terminal.

Ohio Scientific's 540 Video Display board, an assembled PC board from its Challenger IIP personal computer, features formatting of the display from 32 x 64 to 32 x 32 which allows square cells and hence video animation. The 540 also optionally supports a graphics character generator which features lower case and about 170 special characters for plots and games.

Computer Data Systems offers Versatile 3B and Versatile 4 complete computer systems that fit snugly into terminal enclosures. The Versatile line uses the 8085 CPU and a Micropolis disk operating system along with 24K of static RAM. These systems are sleak, compact and well-designed.

Gimix video board for the S-50 bus follows the tradition of this young computer-oriented company – it's a versatile, but unassuming video board. This board works with multiple monitors or on a master antenna system so you can read the display at multiple locations. You can also use more than one board per system. The fully interlaced EIA video output is crystal controlled.

National Multiples' terminal, the TVDM, serves as a terminal with optional tape drives. This three-part system consists of a keyboard, memory and driver PC card, and high resolution monitor, all for the S-100 bus. The tape drives, from 3M, make handy adjuncts to this system.

Telpar recently introduced a full duplex bit serial terminal including a thermal printer. The T-48C is ASCII encoded and operates interactively with the KB-59C keyboard, which is formatted for both Teletype and typewriter capability. This silent, compact unit produces crisp images but is not excessively tiny.

The Multiterm T4000 from Terminal Systems uses the Diablo HyType II mechanism known for its unique graphic capability, high speed tabbing and backward printing under program control.

Soroc Technology's answer to a capable, low cost remote video display terminal is their IQ 120. In addition to a 1920 character screen memory, the terminal features expansion options including block mode and hard copy capability with printer interface.

Electrolabs named their unit "Dumb Terminal for Smart People". According to Electrolabs, this terminal is identical in capability to ADM3 and "then some". It's the only terminal using the 8th bit of ASCII code as a parity bit.

The H9 Video Terminal from Heath offers a long-form display (twelve 80-character lines), a short-form display (forty-eight 20-character lines in four columns) and a plot mode for curves, graphs and simple figures. This terminal works with the Heathkit H8 or H11 computers as well as with any stand-alone or timesharing computer.

CMC Marketing offers an extensive line of terminal computers which they call Processor Terminals. Note the quantity and size of the built-in disk drives for each unit. The start up "jump to" routine is a fast way to put your terminal on line to the 32K RAM 8080 CPU. All models are housed in heavy-duty aluminum cabinets with power supplies, making brownouts a thing of the past.

The CT-64 Terminal from Southwest Technical Products is a kit featuring scrolling or page mode operation, control character printing, complete control character decoding, and other features that make it a video monitor providing everything needed for a complete stand-alone terminal system compatible with modems and ASCII computer systems of every kind.

IMSAI's Video Data Processor, the VDP-80, is a fully integrated twin floppy disk-based system in a single cabinet with 32K memory expandible to 196K.

(Continued on p. 73)

Manufacturer Information – Chart 1

No.	Company Name	Product Name and Number	Avail. From	Avail-ability	Retail Price	Interface Requirements
1	CMC Marketing Corp.	MCS-PT412	RE	1	$8250	NA
2	CMC Marketing Corp.	MCS-PT408	RE	1	6795	NA
3	CMC Marketing Corp.	MCS-PT312	RE	1	6495	NA
4	CMC Marketing Corp.	MCS-PT212	RE	1	5995	NA
5	CMC Marketing Corp.	MCS-PT208	RE	1	4695	NA
6	CMC Marketing Corp.	MCS-PT112	RE	1	4395	NA
7	Computer Data Systems	Versatile 3B; Versatile 4	DS	1	3295/3995	NA
8	Computer Transceiver S.	Execuport 4000	MO,RP	2	–	RS232;20mA c.l.; accous. coup.
9	Datamedia Corp.	Elite 3052A	–	–	–	RS232;20mA c.l.
10	Dataview, Inc.	Marquis	RE,RP	1	805	RS232;20mA c.l.
11	Dynabyte	Naked Terminal	RE	1	350	S100 bus
12	Economy Terminals	ET1/E,ET2,ET2/D	All 4	2	<1500	RS232;20mA c.l.
13	Electrolabs	Dumb Term'l for Smart P'ple	MO	–	329	ASCII code
14	Electrolabs	–	MO	3	495-2000	–
15	GIMIX, Inc.	Video Board	RE	1	198	S50 bus
16	Heath Co.	H9 Video Terminal	MO	1	530kit...	RS232;20mA c.l.; std. TTL
17	IMSAI Mfg. Corp.	Video Data Proc. VDP-80	RE,MO,DS	2	6995	NA
18	IMSAI Mfg. Corp.	Pers'l Comp'g Sys. PCS-80/30	RE,MO,DS	2	1349k	1449a] NA
19	IMSAI Mfg. Corp.	Video Data Proc. VDP-40	RE,MO,DS	2	–	NA
20	Leigh	series 800 teleprinters	(MO)	–	–	8 level 11-bit binary DC code
21	Micro-Term, Inc.	Act IV A + B	RE,MO	2	550;800	RS232
22	Micro-Term, Inc.	Act I A; Act II A	RE,MO	2	400;550	RS232
23	Nat'l Multiplex Corp.	TVDM	RE,NO,DC,RP	1	630	S100 Bus
24	Ohio Scientific	540 Video Displan	RE,MO,DS,RP	–	249	OSI Bus
25	Omni Systems Assoc.	Porta Com	MO,RE	–	595 u 1600 n	Bell 103 Modem
26	Problem Solver Sys., Inc.	TLC	RE	1	995	RS232
27	Processor Tech. Corp.	Sol-20/16	RE	1		
28	Ramsey Electronics	TH3216; TH6416	MO	1	149.95-189.95	TTL-7 unit ASCII
29	Soroc Tech., Corp.	IQ 125	RE,DS,RP	2	1095	RS232 and/or 20mA c.l.
30	Soroc Tech., Corp.	IQ 120	RE,DS,RP	2	995	RS232 and/or 20mA c.l.
31	S.W. Tech. Prod. Corp.	CT-64 Terminal	RE,MO	1	500	RS232 Serial
32	Space Byte S	SB85-16 term.mtd.mainframe	RE,RP	1	5900	NA
33	Telecommunications Serv.	Datapoint 3000/3300/3360	RE,MO	1	649.50 up	RS232
34	Teletype Corp.	Model 42 Product Line	(RP)	–	1000-4000	20/60mA c.l. or telex neu.line intfc.
35	Teletype Corp.	Model 43 Product Line	(RP)	–	1000-4000	RS232
36	Telpari, Inc.	PS-48C interactive ther.prtr.	(MO)	–	888	TTL PLLL, TTL SRL, RS232, 20mA c.l.
37	Terminal Sys., Inc.	Multiterm T4000	RE,DS,RP	–	3095	RS232 and 20mA c.l.

Terminals – Chart 2

No.	Pr/ CRT	Intel./ Dumb	S-C Modem	Editing Capability	S-C FDS	Int.Pt. /Comp.	Scroll Versa.	Scroll Method	Gr. Disp. Resolutn	Mapping Technique	Baud Rate	Features (see footnotes)
1	CRT	8080A	no	line	4x8"	yes	fwd.	line	80x24	–	75-19.2K	1,2,3,4;½+F
2	CRT	8080A	no	line	[3]	yes	fwd.	line	80x24	–	75-19.2K	1,2,3,4;½+F
3	CRT	8080A	no	line	[4]	yes	fwd.	line	80x24	–	75-19.2K	1,2,3,4;½+F
4	CRT	8080A	no	line	2x8"	yes	fwd.	line	80x24	–	75-19.2K	1,2,3,4;½+F
5	CRT	8080A	no	line	2x5"	yes	fwd.	line	80x24	–	75-19.2K	1,2,3,4;½+F
6	CRT	8080A	no	line	5¼"	yes	fwd.	line	80x24	–	75-19.2K	1,2,3,4;½+F
7	CRT	8085	no	line	yes	yes	fwd.	line	80x24	char.ROM	75-76K sync	1;½+F;+
8	Prntr	6800	yes	no	no	no	fwd.	line	136x...	–	110-300	+
9	CRT	dumb	no	–	no	no	–	–	80x24	–	to9600	+
10	CRT	dumb	no	no	no	no	fwd.	line	80x24	–	50-9600	½+F
11	CRT	dumb	no	line	no	(yes)	fwd.	line	NA	char.ROM	300-12K	1;½+F;+
12	CRT	6502	no	line,block	no	(no)	fwd.	line,scan	80x24	–	50-19.2K	1,4;½+F;+
13	CRT	dumb	no	–	no	no	(fwd.)	line	–	–	110-19.2K	1,2;½+F;+
14	CRT	intel.	yes	line,block	8"	no	f + r	line,scan	132x128	bit map	150-19.2K	1,2,3,4,5;½+F;+
15	CRT	–	no	(sftwre)	no	yes	f + r	–	64x16	–	–	–
16	CRT	dumb	no	line	no	no	fwd.	line	80x12	char.ROM	110-9600	½+F;+
17	CRT+	8085	no	line	2x8"	yes	fwd.	line	80x24	char.ROM	.5-9600[5]	1,4;½+F
18	CRT+	8085	no	line	no	yes	fwd.	line	80x24	char.ROM	.5-9600[5]	1,4;½+F
19	CRT+	8085	no	line	2x5"	yes	fwd.	line	80x24	char.ROM	.5-9600[5]	–
20	Prntr	dumb	yes	–	no	no	fwd.	line	–	–	110	–
21	CRT	F8	no	line,block	no	no	fwd.	line	80x24	char.ROM	110-19.2K	3;½+F
22	CRT	dumb	IIA only	line	no	no	fwd.	line	80x24	char.ROM	110-19.2K	3;½+F;+
23	CRT	-80/Z80	no	(sftwre)	op.[6]	yes	f + r	scan	80x24	char.ROM	110-19.2K	1
24	CRT	dumb	no	(sftwre)	no	yes	f + r	–	64x32	–	–	+
25	Prntr	dumb	yes	–	no	no	fwd.	line	80x...	–	16.3K	–
26	CRT	Z80	no	line,block	nxt[7]	no	f + r	line,scan	80x24	char.ROM	75-19.2K	1,2,3,4;+
27	–	–	–	–	–	–	–	–	–	–	–	–
28	CRT	dumb	no	line	no	no	f + r	line	64x16	char.ROM	110-9600	+
29	CRT	(dumb)	no	line,block	no	no	fwd.	line	80x24	–	110-19.7K	1,2,3,4;½+F
30	CRT	dumb	no	line,block	no	no	fwd.	line	80x24	char.ROM	75-19.2K	3,4;¼+F
31	CRT	dumb	no	no	no	no	fwd.	line	64x16	–	110-1200	1;½+F;+
32	CRT	8085	no	line	2x8"	yes	f + r	line	80x24	–	110-19.2K	3;½+F;+
33	CRT	7400s[8]	no	line	no	–	f + r	line	82x25	char.ROM	110-9600	½+F;+
34	Prntr	intel.	no	no	no	no	(fwd.)	line	132x...	–	45.5-225	–
35	Prntr	dumb	op.	no	no	no	(fwd.)	line	132x...	–	110-1800	+
36	Prntr	F8/3870	no	–	no	no	–	line	48x...	–	110or300	F;+
37	Prntr	8080	no	no	no	no	f + r	line	–	graphics	110-1200	+

CRTs – Chart 3

No.	Type/ CRT	Diag.(") Scr. Sz.	Colors Avail.	Max. Char. Display	Hot or Isolated	Audio Capability	Char. Matrix	P-PC VIV	MHz Band.	Ohm Imp.	Power Requirements
1	mon.	15	b/w	80x24	isolated	–	5x7	1.	22	75	110VAC
2	mon.	9	b/w	80x24	isolated	–	5x7	1.	22	75	110VAC
3	mon.	15	b/w	80x24	isolated	–	5x7	1.	22	75	110VAC
4	mon.	15	b/w	80x24	isolated	–	5x7	1.	22	75	110VAC
5	mon.	9	b/w	80x24	isolated	–	5x7	1.	22	75	110VAC
6	mon.	15	b/w	80x24	isolated	–	5x7	1.	22	75	110VAC
7	mon.	9	b/w	80x24	–	–	5x7	–	–	–	–
8	NA	–	–	–	–	–	–	–	–	–	–
9	mon.	9	b/w	80x24	(isolated)	(none)	–	–	–	–	–
10	NA	NA	b/w	80x24	NA	NA	5x9	–	NA	NA	NA
11	NA	NA	–	80x24	NA	NA	–	–	NA	NA	NA
12	mon.	12	P4	80x24	isolated	none	7x9[1]	–	[2]	–	110VAC;25w
13	–	–	–	–	–	–	–	–	–	–	–
14	–	–	–	–	–	–	–	–	–	–	–
15	–	–	–	64x16	–	–	–	–	–	–	–
16	mon.	12	b/w	80x12	isolated	none	5x7	–	–	–	110VAC
17	mon.	12	b/w	80x24	isolated	none	7x10	1.5	12	75	110/220VAC;200w
18	mon.	5	b/w	80x24	isolated	none	7x10	1.5	14	75	110/220VAC;50w
19	mon.	9	b/w	80x24	isolated	none	7x10	1.5	14	75	110/220VAC;120w
20	NA	–	–	–	–	–	–	–	–	–	–
21	mon.	12	b/w	80x24	isolated	none	5x7	–	–	–	115VAC
22	–	–	b/w	80x24	–	–	5x9	–	–	–	–
23	mod.TV	12	b/w	80x24	isolated	std.amp/spkr	7x9	2	10	75	110VAC;150w
24	mon.	–	b/w	64x32	–	–	5x7	–	–	–	–
25	NA	–	–	–	–	–	–	–	–	–	–
26	–	–	–	–	–	–	–	–	–	–	–
27	–	–	–	–	–	–	–	–	–	–	–
28	NA	NA	b/w	64x16	NA	NA	5x7	NA	NA	75	NA
29	mon.	12	(b/w)	80x24	isolated	(none)	5x9	–	–	–	115/230VAC
30	mon.	12	(b/w)	80x24	isolated	std.amp/spkr	5x7	–	12	–	–
31	mon.	9	green	64x16	isolated	output jack	6x9	2.5	12	500	12VDC;10w
32	mon.	12	b/w	80x24	isolated	(bell)	–	–	–	–	–
33	mon.	12	P31	82x25	isolated	sgnl tone spkr	5x7	–	–	–	117/230VAC
34	NA	–	–	–	–	–	–	–	–	–	–
35	NA	–	–	–	–	–	–	–	–	–	–
36	NA	–	–	48x...	–	–	5x7	–	–	–	–
37	NA	–	–	–	–	–	–	–	–	–	–

Keyboards – Chart 4

No.	How Encoded	Keyboard Configuration	Voltage Requirements	N/P Strobe	Data Lines	Asm./ Kit	Type of Keyboard Mechanism
1	ASCII	Standard	from unit	–	–	asm.	High-Tech
2	ASCII	Standard	from unit	–	–	asm.	High-Tech
3	ASCII	Standard	from unit	–	–	asm.	High-Tech
4	ASCII	Standard	from unit	–	–	asm.	High-Tech
5	ASCII	Standard	from unit	–	–	asm.	High-Tech
6	ASCII	Standard	from unit	–	–	asm.	High-Tech
7	ASCII	U+LC alpha+num pad	110VAC;400w	positive	8	asm.	reed switch keyboard
8	128c.ASCII	TTY/APL(swtchbl)	110-130VAC	sync	–	asm.	bit pad; typewriter pad
9	ASCII	Standard	–	–	8	asm.	–
10	64c.ASCII	Standard	115-130VAC	–	11	asm.	–
11	NA	–	–	–	–	–	–
12	128c.ASCII	Standard	110VAC;25w	–	–	asm.	contact style switches
13	–	–	–	–	–	–	–
14	ASCII	–	–	–	–	–	–
15	ASCII	–	8VDC@.7A[9]	–	8	asm.	–
16	67c.ASCII	Standard	110VAC	–	–	kit	–
17	ASCII	Standard	main power sup'ly	negative	8	asm.	mechanical
18	ASCII	Standard	main power sup'ly	negative	8	asm./kit	mechanical
19	ASCII	Standard	main power sup'ly	negative	8	asm.	mechanical
20	ASCII	Standard	–	–	11	asm.	–
21	128c.ASCII	TTY;Std.+num pad	–	–	–	asm.	–
22	128c.ASCII	TTY; Std.	–	–	–	asm.	–
23	ASCII	Standard	+5VDC	neg.+pos.	–	asm.	solid state switch
24	ASCII	Standard +	–	–	–	asm.	–
25	64c.ASCII	Standard + num pad	115VAC	–	–	asm.	–
26	64c.ASCII	TTY ASCII Std.	105-130VAC	–	–	–	–
27	128c.ASCII	–	–	–	–	–	–
28	64c.ASCII	–	+5VDC@1A max	–	11	asm./kit	NA
29	ASCII	Standard	+5VDC	negative	8	asm.	mechanical
30	ASCII	–	–	positive	7	asm.	mechanical switch
31	ASCII	Standard	+5,-12VDC	either	8	asm./kit	mechanical keyswitches
32	ASCII	Standard	110VAC	–	–	asm.	–
33	ASCII	Standard	117/230VAC	–	–	asm.	magnetic key switches;diode matrix;n-key r/o
34	50c.CCITT 2	TTY/Typewr+num pad	104-127VAC;125w	–	–	asm.	Hall effect
35	ASCII	TTY/Typewriter	115VAC; 205w	–	–	asm.	Hall effect
36	ASCII	TTY/Typewriter	–	–	–	asm.	–
37	APL	Selectric (IBM)	110VAC@3A;120w	–	–	asm.	–

CRTs

Manufacturer Information – Chart 5

No.	Company name	Product Name & Number	Avail. From	Avail-ability	Retail Price	Interface Requirements
1	AEG-Telefunken Corp.	D5-100 CRT	MO	1	48.	Tube only. Req. electronics +
2	Computer Warehouse	Video Monitor	MO	1	150. +	
3	Electrolabs	Sylvania Monitor (used) 17"	MO	1	24.95 up	Composite Video
4	Image-21	IMV-12A Monitor	MO	1	---	Composite Video
5	Panasonic	140AYB31 CRT	MO	3	---	Tube only. Req. electronics +
6	Telecom. Services Co.	Datapoint 3000/3300/3360	RE,MO	1	649.50+	RS232(20ma pr.output only avail)

CRTs – Chart 6

No.	Type/ CRT	Diag.(") Scr. Sz.	Colors Avail.	Max char Display	Hot or Isoltd	Audio Cap.	Char. Matrix	P-PC V/V	MHz Bandw.	Ohm inp.	Power Req.
1	(CRT)	2"	b/w	---	---	---	625 lines res.	---	---	---	---
2	Mon.	12"	P39 grn	80x24	150.	None	7x9	1.4	16	50	115VAC;60W
3	Mon.										
4	Mon	9"-17"	b/w	---	---	---	600-800 LI res.	---	---	---	---
5	(CRT)	5"	b/w	(int.grat)	---	---	---	---	---	---	---
6	Mon.	12"	b/w	80x24+	150.	---	---	2	15	75	110VAC
1	UHF tv	any	b/w-col	---	(150)	Audio	---	---	---	---	+5,+1V/bat.
2	any tv	any	b/w-col	---	(150)	4.5MHz[1]	---	---	---	15V/50ma	2ma MAK
3	any tv	any		---	(150)	4.5MHz[2]	---	.25-5V	---	---	35W
4	int.mod.	9"or12"	b/w	96x	150.	---	---	.6-4V	>10	75 or 5K	117AC

Special Devices Re: CRTs

No.	Company name	Product Name & Number	Avail. From	Avail-ability	Retail Price	Interface Requirements
1	ATV Research	Micro-Verter MVX500	MO,RE	1	35.	RF coup.dir.into UHF tuner input+
2	ATV Research	Pixe-Plexer PXP4500	MO,RE	1	24.50	Std. compos. video +
3	ATV Research	Pixe-Verter PXV-2A	MO,RE	1	8.50	Std. compos. video: <3 ma
4	Pickles & Trout	TVM-41 TV Mod Kit	RE,MO	1	20.	Std. compos. video +

KEYBOARDS

Manufacturer Information – Chart 7

No.	Company name	Product Name & Number	Avail. From	Avail-ability	Retail Price	Interface Requirements
1	Cherry Elec. Prod. Corp.	"PRO" B70-05AB	MO,DS,RP	1	135.*	TTL output positive logic
2	C.P. Clare & Co.	(Solid state keyboard)	MO (+)	---	---	SRL and/or PLLL I/O
3	Data Interfaces	KSA-35 keyboard	MO	---	---	---
4	The Digital Group	Key/Comp-keyboard	RE, MO	2	245.	8 bit TTL/ASCII PLLL
5	EECO	Programmable Keybd.	RE,MO	---	---	---
6	Electrolabs	Cherry "Pro"	MO	1	119.ea-3/99.	8 bit ASCII PLLL
7	Geo. Risk Indust.	Model 716 HexPad	RE,MO,DI,RP	1	29.95K	4 bit Hex + STR
8	Geo. Risk Indust.	Model 753	RE,MO,DI,RP	1	59.95K[1]	Std. ASCII PLLL+strobe output[2]
9	Geo. Risk Indust.	Model 756 full ASCII kbd.	RE,MO,DI,RP	1	150.	7 bit ASCII PLLL+strobe output[3]
10	Geo. Risk Indust.	Model 771	RE,MO,DI,RP	1	150.	7 bit ASCII PLLL+strobe output[4]
11	MicroAge	MKB-2 keyboard	MO(+)	---	149.	PLLL, strobe or pulse
12	Static Systems Corp.	Liq. Crys. al.num. display	---	---	---	(SRL)

Keyboards – Chart 8

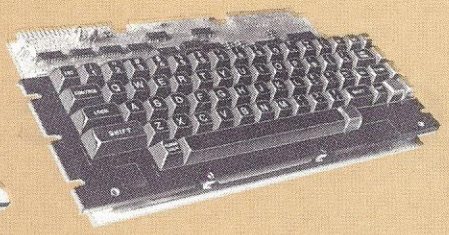
No.	How Encoded	Keyboard Configuration	Voltage Requirements	N/P Strobe	Data Lines	Asm./ Kit...	Type of Keyboard Mechanism
1	ASCII	Standard	+5VDC	N and P	8	asm.	gold cross-point
2	(optional)	(Standard)	–	–	–	asm.	capacitive keyswitch
3	ASR-33 TTY	Std. + num pad	–	–	–	asm.	gold cross-point
4	ASCII	ASCII	+5VDC@1A max	Positive	7	asm.	capacitive keyswitch
5	(optional)	(Standard)	–	–	–	asm.	–
6	ASCII/TTY	Standard	+5VDC@.35A	neg or pos	8	asm.	gold cross-point
7	Hex	Hex (4x4)	+5VDC@30mA	N or P	4	asm./kit	gold cross-point
8	ASCII	ASCII	+5VDC@15mA[1]	N or P	8	asm./kit	double gold cross-point mech sw.
9	ASCII	ASCII	+5VDC@15mA[2]	N or P	8	asm./kit	double gold cross-point mech sw.
10	ASCII/TTY	ASCII/TTY	110VAC	N or P	8	asm.	double gold cross-point mech sw.
11	ASCII	(Std.) + num pad	+5VDC, +12VDC	–	–	asm.	–
12	"Selectric"	Standard	+5VDC;15μw	–	–	asm.	Hall effect solid state keys

FOOTNOTES

1–or 5x7; 2–15.7 or 18.972 mHz; 3–2x8" and 2x5¼"; 4–2x8" and 1x5¼"; 5–async/ 30.5-56K sync; 6–also tape drives available; 7–to be included in the next model; 8–semi-intelligent using the 7400 series; 9–and -12VDC@.04A.

Blank spaces in these charts result from lack of adequate information from the manufacturer or supplier.

IMSAI's Video Data Processor VDP-80/1000



Cherry Electrical Products' "PRO" Keyboard

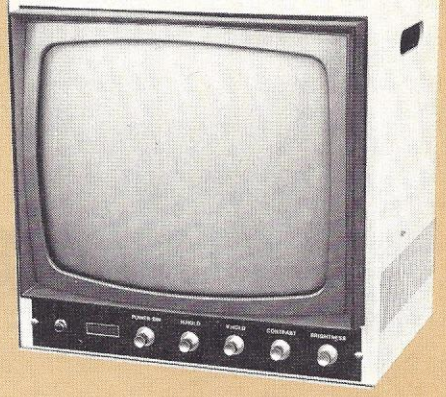


Image-21's Video Monitor IMV-12A

(Continued from p. 69)

Additional floppy disks can provide more than 5 megabytes of additional data.

The VDP-40 and the Personal Computing System (PCS-80/30) are two additional terminal/computers manufactured by IMSAI. The VDP-40 is similar to the VDP-80, but differs in the size of the CRT and the floppy disk drives. The PCS-80/30 also provides sophisticated terminal functions along with the microcomputer operations.

Telecommunications Services offers completely refurbished Datapoint CRT terminals at about one-fifth (and up) the original price. Each unit comes with a 90-day guarantee, an excellent appearance, new faceplate and (if necessary) repainted cabinet. Check the features on these terminals. They are "used" but still have a long life to offer you.

Economy Terminals lists their ET1/E, ET2 and ET2/D terminals, which include controls for brightness, DIP switches for parity, stop bits and editing modes. These 6502-controlled terminals offer a bundle of features packaged into an economical system.

Microterm markets four different models: Act-IA, Act-IVA and B, and Act-IIA. All four reasonably priced units are professional in appearance and have had excellent user response. The manufacturers say that even IBM is a repeat customer!

The Naked Terminal (VT801-1) by Dynabyte allows for status bit to be either positive true or negative true and may be assigned to any bit of the status word. The status port may be addressed anywhere. According to the manufacturer, no support software is required.

The Marquis, marketed by Dataview, is a truly quiet terminal, with neither fans, nor other objectionable noise features. The two-year warranty offers extensions at a small charge.

Ramsey Electronics offers the Cybernex terminal card kits. Both models TH3216 and TH6416 offer full cursor control, complete stand-alone operations, UART included on-board and sockets for all ICs. The TH6416 has the sync brought out to a connector for external uses such as titling. All timing is syncronous-derived from one crystal, using +5V DC only.

Problem Solver Systems introduced their TLC which includes 16 fully programmable function keys. There is space on board for 8K EPROM and 8K RAM as well as a printer output port.

Computer Transceiver Systems recently announced a low-cost, 16-pound send/receive portable terminal with a full 136 columns. This terminal, the Execuport 4000, provides ASCII, APL or IBM PTTC/EBCD codes with operating speeds up to 30 cps. Many new features, including true upper and lower case printing, tabbing, built-in testing, print position display, buffered carriage return and parity error detection, are incorporated into the terminal.

Cherry Electrical Products' Pro B70-05AB keyboard features five unassigned re-legendable keys. The unique alpha lock key changes the keyboard from typewriter to teletypewriter code outputs. All keys are hard-wire programmable, and and customizing is easy.

George Risk Industries offers several keyboard kits for professionals as well as hobbyists. Model 771 includes a wrinkle finish desk top all-steel enclosure and a standard "D" series 25-pin interface connector. Model 753 offers three user-defined keys for custom programming. Model 756 is based on either positive or negative logic, uses gold contact, low bounce and according to the manufacturer, was specifically designed for use with all microcomputer systems. Check out the variations in the specs on the chart.

Pickles and Trout's TVM-41 TV modification kit (for direct video input) is designed primarily for Hitachi model TVs and includes a 5-foot video cord and a tool for adjustment of picture height and width.

ATV Research offerings include three very simple, yet high quality devices for RF modulated input to standard TV sets. The Pixe-verter, a transistorized modulated oscillator, works on all TV cameras, VTRs, computers, video games and so forth. It accepts digital or analog video from 0.25 to 5V DC. Wide bandpass allows color and 4.5 MHz optional sound subcarrier operation.

The Micro-verter, also by ATV Research, is designed to interface microcomputers to color or monochrome television by operating in the UHF channels above channel 14, beyond the normal range of switching harmonics which interfere with video quality on the VHF channels.

Pixe-Plexer is a more sophisticated Pixeverter designed for multiplexing and interfacing color and luminance video signals plus audio for display on any regular TV via the antenna terminals.

Image-21, a division of ATV Research, deals with CRT monitors for computer and CCTV applications. Screen sizes range from 9" to 15" and resolution from 600 to 800 lines. Each unit carries a year's warranty on parts and service. Sharp-edge resolution for easy character recognition is one of the main features of these monitors.

Now that you have an overview of terminals, CRTs and keyboards, you can assess your own requirements and decide what best fulfills your needs — either from our lists, from your local computer retailer or from mail order businesses. Hundreds of potential terminals and terminal systems could be listed, but for the sake of space and efficiency we've sampled only a few of the largest sources with the best reputations. We hope these listings will help you find the unit you need in addition to giving you practical buying information. □

FIRMWARE FACTS

Voltage on the Level

BY AUSTIN LESEA AND RODNEY ZAKS

In my wanderings forward I have found many frustrated computer hobbyists wandering backward. Hardware people do not know a compiler from an interpreter. Software people pick up the wrong end of the soldering iron. The solution here is obvious: join forces, friends. Become firmware people, computer designers – people who can master the technology, and not be mastered by it.

In an effort to get designers thinking along these lines, I have prepared this column on a design area that baffles the software people, as well as bothering hardware people.

How do you take what comes out of the wall and transform it into +5.00 volts at 10.00 amperes? How do you do it economically, without endangering your computer's health and well being? Let us start at the beginning.

Initially, we have 117 volts RMS at 60 hertz coming from the wall socket. What does that mean, exactly? The stated voltage is the value of the pressure or force electrons have at the contacts of the socket. The RMS indicates Root Mean Square, which is the average voltage present. The 60 hertz indicates that the voltage changes from positive to negative and back again 60 times per second.

The shape of the voltage versus time is that of a sine wave. The peak value of this sine wave is 1.414 times the RMS value and for the 117 volt wall socket it is about 165 volts. The peak to peak value is twice the peak value, or 330 volts peak to peak.

Usually the wall socket will have a current rating also. Electrical current measured in amperes is the quantity of electrons which flow by a given point in one second. Larger currents require physically larger wires so the current flow will not be impeded.

Product sections, with their many undefined electronics terms, sometimes leave hobbyists confused. Such terms as 2v at 255 amps, 100/117 VAC at 60 Hz, peak values, RMS, 70 VA operating and heat sinks, have meanings not immediately clear. Many, on occasion, even send you to a text book – if you can find one readily.

Personal Computing plans to bring you reports of this kind, dealing with the language of electricity and electronics, to better help you understand and select computer hardware without having to become an electrical or digital engineer.

Typical wall sockets have wires large enough to safely carry 20 amperes. If the current flow becomes impeded by a resistance, energy is wasted in the form of heat. The amount of energy lost is related to the voltage drop across the resistance times the current that is flowing through it. Power is measured in watts and equals the product of volts times amps. In fact, because a computer generates no mechanical work, no light, and performs no chemical reactions, the computer is really nothing more than a heater. It takes power from the wall socket, and uses it to rearrange numbers and perform computations. Remember, when things begin to warm up power is being dissipated.

Now, at the other end, we have circuit boards of logic which demand various voltages at various currents. In the simplest case, we require +5.0 volts ±5%, at 3 amperes, DC. That does not sound at all like the wall socket. In fact, we have to transform the wall voltage supply into a suitable supply before we can do anything. Let us analyze the requirements.

The 5.0 volts already mentioned is the pressure or voltage that our machine needs. The maximum current that can be drawn is 3 amperes. This means that our supply will deliver 15 watts of power to the circuits, at the maximum. The ±5% tolerance tells us that the voltage must be between 4.75 and 5.25 volts. Out of this range, the circuit will not work. DC is direct current and the wall socket AC or alternating current. The 5 volts is a constant voltage that never changes polarity, (which means that the electrons flow in only one direction).

To use our wall socket we must reduce its voltage, change it to DC, and regulate it to within the required tolerance. This is transforming, rectifing, and regulating, respectively.

Transforming the voltage means we are going to step it down to a suitable value. General supply-design rules dictate that this voltage should be twice the regulated output after rectification. The rectifiers will deliver the peak value of the reduced AC voltage, so we must divide the desired value by 1.414. In this case, we would like 10 volts unregulated DC before the regulator. Dividing by 1.414 we get 7.07 volts RMS. Thus we need a transformer that will shrink 117 volts down to 7 volts. The current required is 3 amperes at 5 volts. We will still draw 3 amperes from the 7 volts of the transformer because the

rest of the voltage will be used as margin for the regulator. Power required then at the secondary terminal of the transformer is 21 watts. Transformers are 80% efficient, therefore we will draw 26.25 watts from the wall socket. How many amperes at 117 volts is this? It is 26.25 divided by 117 or 0.22 amperes.

Our transformer is a 26 watt size, with a 117 to 7 volt turns ratio. The output is rated at 3 amperes, RMS.

Now we must change the sine wave at 7 volts into direct current. Using rectifiers which will force electrons to flow in only one direction, we change the AC voltage into a pulsating DC voltage. Basically, by arranging four diodes we can move the lower half of the sine wave up to the top. This waveform then has half sine-waves next to each other, 10 volts peak tall. By placing a capacitor on the output of the diodes, the peaks can be smoothed out to fill the valleys. We now have our unregulated 10 volts DC.

The regulator will do its best to keep 5 volts exactly on target by throwing away, in the form of heat, the unused power. Here, we throw away 5 volts of whatever current is being drawn. At the maximum, this will be somewhat less than 5 volts (due to peak-and-valley filling to about 7 volts DC under full load) times 3 amperes. At full load this is 2 times 3 or 6 watts. Thus, if the input to the regulator falls below 7 volts, which here will be the minimum voltage before the regulator ceases to regulate, the output 5 volts will drop also. To insure against this problem, the AC stepped down voltage is normally specified at 10% less than the doubled value of the required output voltage. If the wall socket goes above 117 volts, we will just throw away more heat in the regulator to keep the output constant. If the wall socket voltage drops, we can regulate until about 90 volts.

Now we have to choose the regulator. We can either design our own from discrete components, or find a monolithic device. In our case there exists the LM323 three ampere 5 volt regulator chip. We connect the unregulated 7 to 10 volts into the IN pin, and 5 volts +5% appears on the OUT pin up to a current level of 3 amperes. If any larger current is drawn, or the regulator gets too hot, it will automatically shut down without damage.

The regulator package must be carefully mounted on a suitable heatsink. At the worst, we must dissipate 18 watts. (This is 10% high line voltage with maximum current being required on output.) An 18 watt heatsink is about 4" by 6" with 8 one-inch high fins, in still air.

Besides heat sinking, caution is necessary in the layout to avoid oscillation or instability. The input is bypassed with a 0.1 microfarad capacitor. The output is tested similarly. The capacitors are mounted less than one inch from the regulator.

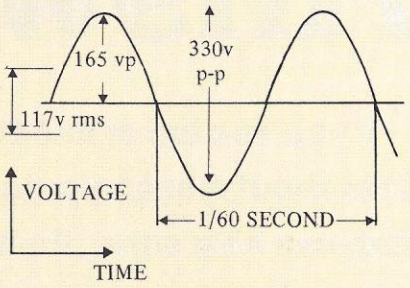
You must never increase these values to solve an oscillation problem. Higher value capacitors have too much internal inductance to function effectively. You will actually make it worse. If a problem exists, check layout, grounding, and the bypass capacitors. High quality ceramic or tantalum capacitors are required, which have low internal inductance.

Transforming, rectifying, filtering, and regulating are the basic requirements discussed. At each step many choices have been made. In this article most steps were simplified. The real experts in this field are the O.E.M. suppliers (original equipment manufacturers.)

O.E.M. power supplies are manufactured in quantities of hundred thousands. Cost in one or two quantities runs about a dollar per watt. The supply reflects the amount of engineering money spent and the power unit will probably never go bad if specified properly. Most minicomputer and microcomputer manufacturers start with healthy O.E.M. power supplies rather than risk the company's reputation on bad power supply design techniques.

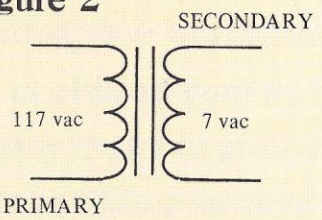
Power supplies are not simple. In fact, the design of a good power supply will cost as much as the rest of the system, if done properly. □

Figure 1



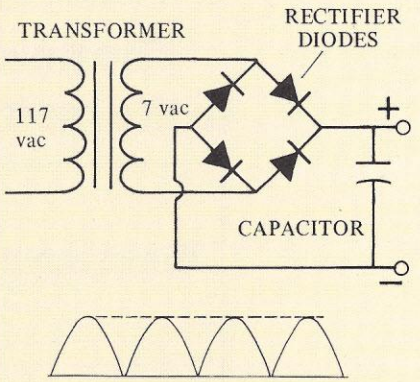
AC line voltage: a sine wave with 165 volt peaks. Time between peaks is 1/60 of a second.

Figure 2



A transformer: for stepping the AC line voltage down or up to another value. (In this case we are going from 117 down to 7.) The ratio of input to output depends on the number of turns of wire on the primary and secondary.

Figure 3



Full wave bridge rectifier: the lower half of the AC voltage is "flipped" over to the top by commutation of the diodes. The filter capacitor smooths this pulsating current into "raw D.C."

Get Some Answers

If you're a top or middle manager who knows little about minicomputers and the problems they can help solve, then you're just the person we can help.

Twice a month — in easy-to-read, easy-to-understand articles — you'll discover how your colleagues in business, industry and science use — or sometimes abandon — minicomputer solutions to problems.

You may be able to apply their solutions directly to your problems, use their solutions to spark your own ideas or you may develop solutions wholly your own.

No other publication gives you as much know-how about minicomputers and how they help solve business problems as MINICOMPUTER NEWS — and all for just $9.00/year. Just fill in the coupon below and we'll start helping you right away.

Minicomputer News

The newspaper that gives real coverage to the rapidly growing minicomputer market.

RATES	1 yr.	2 yrs.	3 yrs.
U.S.	$9	$15	$18
Canada & Mexico	$15	$25	$30
Other foreign —			
Surface	$40	$70	$80
Air	$60	$100	$120

Minicomputer News
1050 Commonwealth Ave., Boston, MA 02215

Name_____ Title_____
Company_____ Zip_____
Address_____
City_____ State_____ Zip_____
Outside U.S.A.: Country Name_____

I buy/specify (circle all that apply)
1. Minicomputers
2. Small business systems
3. Intelligent terminals
4. Microcomputers/processors
5. Add-on memories
6. Disk/tape drives
7. Printers/plotters
8. Interactive terminals (CRT/hardcopy)
9. Punched tape/card/OCR equipment
10. Modems/multiplexers
11. Software/support services
12. Word processors
13. Work stations/enclosures/secured systems
14. DP supplies

My business/industry (circle one):
1. Finance/insurance/real estate
2. Wholesale/retail trade
3. Industrial/consumer products mfr.
4. Utilities/communications/transport
5. EDP services
6. Health care/education/law
7. Printing/publishing/word processing
8. Petroleum/chemical/mining/construction
9. Government/military
10. Minicomputer/peripheral manufacturer
11. Turnkey systems/software
12. Consulting (EDP)
13. Research

My job function/title (circle one):
1. General/corporate management
2. Financial management
3. DP management
4. Engineering management
5. Marketing management
6. Systems/applications eng.
7. Consultant
8. Purchasing/procurement
9. Sales/distribution

A. Education/Library
Z. Other

NOTE: The information in the buy/use, business/industry and function/title columns is needed for statistical purposes so that we know who our readers are and what products interest them.

To order your subscription, clip this coupon and mail it with your check or money order. PC/7

Computer Chess Programming

BY MIKE VALENTI

Following is a summary of steps you must take in developing a computer chess program. Details of recommended programming techniques are not included. (Details may be found in Valenti's thesis running in the computer chess section.) An overall approach of the technique is described. This basic method was used in developing *Chute,* a chess program, on a large IBM system at the University of Toronto.

Basic Rules

Details of some of the legal moves can get fairly involved. Most programmers would miss these details at first glance. This situation can be rather embarrassing at a later date, when you discover your program made an illegal move, especially in a tournament, which is where most program bugs show up.

For instance, in castling, you must make sure that the king is not in check on any of the squares between the king and the rook. But the computer sometimes stumbles past this rule. Another common oversight is moving a piece that is pinned against your own king, thereby exposing your king to an attack, which is illegal.

Internal Representation

You must decide how to store information about the chess board in the computer. For instance, the board itself should be stored as a 10 x 12 array, rather than the most obvious 8 x 8 array. By placing special indicators in the squares outside the 8 x 8 playing area (which is imbedded in the 10 x 12 array as shown in Figure 1), it's easier to check for legal moves for all pieces, including the Knight.

This 10 x 12 board can be simply stored as a vector of dimension 120, and the values in the vector can indicate the type of piece, or empty squares. I used a 12 x 12 board in *Chute*, not realizing that 10 x 12 was sufficient. Figure 1 shows a sample board with numeric values for pieces, empty squares and off-board squares. The values are as follows:

White Pawn	1	Black Pawn	-1
White Bishop	2	Black Bishop	-2
White Knight	3	Black Knight	-3
White Rook	4	Black Rook	-4
White Queen	5	Black Queen	-5
White King	6	Black King	-6
Empty square	0	Off-board sq	99

Input and Output of moves

This is the first step in programming chess. You must enter a move somehow, and display a board. One way to recognize moves is to use a simple notation, such as algebraic notation, where the rows of the chess board are labelled 1 to 8, and the columns A through H. More involved notations could be recognized, but would require much more work, such as P-K4 and N-QR3.

When displaying a board, a number of notations could be used as shown in Figures 2 and 3. In Figure 2, the pieces are indicated by a two-letter code with B for black and W for white, followed by the type of piece (R for rook, P for pawn). White squares are shown by --, and black squares by **.

Figure 3 shows a more abbreviated board, where black pieces are indicated by the minus sign in front of the type.

row →	1	2	3	4	5	6	7	8	9	10
col										
1	99	99	99	99	99	99	99	99	99	99
2	99	99	99	99	99	99	99	99	99	99
3	99	-4	-3	-2	-5	-6	-2	-3	-4	99
4	99	-1	-1	-1	-1	-1	-1	-1	-1	99
5	99	0	0	0	0	0	0	0	0	99
6	99	0	0	0	0	0	0	0	0	99
7	99	0	0	0	0	0	0	0	0	99
8	99	0	0	0	0	0	0	0	0	99
9	99	1	1	1	1	1	1	1	1	99
10	99	4	3	2	5	6	2	3	4	99
11	99	99	99	99	99	99	99	99	99	99
12	99	99	99	99	99	99	99	99	99	99

Figure 1: Internal storage of chess board (at the start of a game).

8	BR	BN	BB	BQ	BK	BB	BN	BR
7	BP	BP	BP	BP	BP	BP	BP	BP
6	--	**	--	**	--	**	--	**
5	**	--	**	--	**	--	**	--
4	--	**	--	**	--	**	--	**
3	**	--	**	--	**	--	**	--
2	WP	WP	WP	WP	WP	WP	WP	WP
1	WR	WN	WB	WQ	WK	WB	WN	WR
	A	B	C	D	E	F	G	H

Figure 2: Sample printing of chess board

8	-R	-N	-B	-Q	-K	-B	-N	-R
7	-P	-P	-P	-P	-P	-P	-P	-P
6	-	*	-	*	-	*	-	*
5	*	-	*	-	*	-	*	-
4	-	*	-	*	-	*	-	*
3	*	-	*	-	*	-	*	-
2	P	P	P	P	P	P	P	P
1	R	N	B	Q	K	B	N	R
	A	B	C	D	E	F	G	H

Figure 3: More compact printing of chess board.

At this stage in writing your chess program, you haven't expended too much effort, but you have something to work with. Namely, you can enter a move and display the chess board. Now things start to get exciting, as you have

established the frame work for communicating with your program.

Legal Move Generation

Here you generate all the legal moves from a given position. You would start first with the simpler pieces (moving pawns straight ahead), then work on the more complicated moves (moving the knights). You also have to worry about special moves, such as en-passant pawn captures and castling. At the end of this phase you should be able to produce a list of all the legal moves from a given position. Careful programming is necessary now.

Chances are you will miss some moves, but these will show up in later testing. In developing *Chute*, a nasty bug showed up several months after I thought it was performing all legal moves. In a position where the king was on the left edge of the board and an enemy pawn was on the right edge of the board but a row above, it thought the king was in check. This failure was due to improper checking for the edge of the board. *Chute* had played in the First Canadian Computer Chess Championships (Waterloo, Ontario, 1974) with this bug, but luckily, the situation did not occur. Later testing uncovered it.

Position Information

You're now getting into the "guts" of the program. This is where you look for various features about a particular position. Starting out, you would check for simple things, such as which squares are being "attacked". If many squares are under attack this could indicate a good position. Then you might check for those enemy pieces that are attacking your own pieces and see if the attack is damaging or not. (You must decide whether you would lose material in an exchange). More complicated situations could be explored, such as pinning enemy pieces.

At this point you must decide how to store this information. For instance, you could use a bit-array (or table) to show which enemy squares can be attacked by your pieces.

Now you must assign numeric values to the various types of pieces (not to be confused with the numeric codes shown in Figure 1). A simple value assignment is:

Pawn	=	1
Bishop	=	3
Knight	=	3
Rook	=	5
Queen	=	9
King	=	99 (for convenience)

This value assignment doesn't consider such things as two bishops being more valuable than two knights.

Move Evaluation

Making the moves one by one, you evaluate them, using the information mentioned in the preceding paragraphs. You make decisions as to whether various features are good or bad. These decisions are usually grouped together into "heuristics" (searching a tree for problem solutions).

For instance, a heuristic for positioning rooks would check to see if it is on an open file, or behind his own passed pawns. By this time, your program would be able to play legal chess and make some reasonable moves (for a computer). However, it would easily get into trouble, as it has no "look-ahead".

The Look Ahead

Here the program makes plausible moves internally and tries to determine how the opponent might respond. A "game tree' would be constructed showing what position you would reach with your move. Figure 4 shows a possible simple game tree for white's move at the beginning of a game (as in Figure 2).

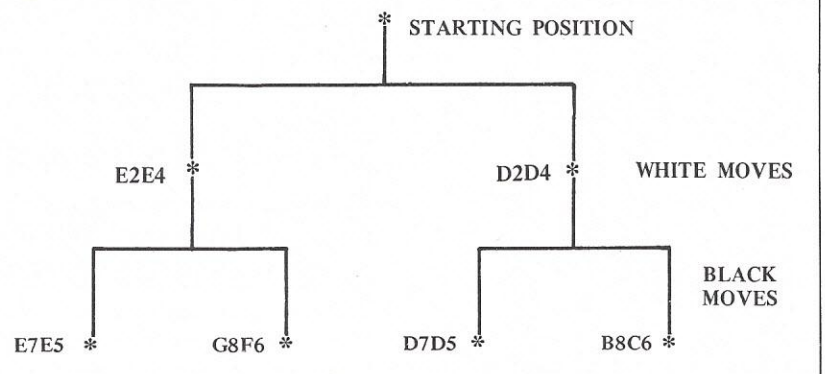


Figure 4: Simple game tree at start of game.

Figure 4 shows how white might make two moves at the start of the game, namely E2E4 or D2D4, and then shows possible responses by the opponent to each of these moves. Scores would be associated with these positions, and the computer would choose the move that would attain the best position. At this stage you have a program which plays a reasonable game of chess and avoids some obvious blunders by using the look-ahead technique.

Book Openings

Most programs require some guidance at the beginning of a game. Just as humans use standard openings, programs can use pre-selected moves from a "book" of recommended openings. Some programs have very extensive books covering thousands of opening possibilities. *Chute* used a different approach, with a very small book (only 40 or 50 variations). Non-standard openings were used by *Chute* to throw other programs out of their books. This approach proved successful when playing computers, but could leave the computer at a disadvantage when playing against a human player.

Other Items

At this stage you could introduce more advanced features into your program, such as tree-pruning.

Alpha-Beta tree-pruning is a technique that saves considerable computer time in the look-ahead step. Basically, if the computer sees a bad position in the look-ahead, this technique prevents further evaluation of the move. For a human player, this is a simple concept; namely, if a move looks bad, you don't spend more time trying to see if it could look worse.

Another useful feature in developing your program is a command language to control the use of the program. This allows you to enter a non-standard starting position, to take back moves (useful in tournaments when you type the wrong move by mistake) and to adjust evaluation parameters.

The steps outlined above are some of the phases I went through in developing *Chute* on a large IBM computer system. For use in a microcomputer, some of the steps would have to be combined or abbreviated due to the limitations of the smaller machine. But the basic concepts are the same. □

Cyber 176 tops Toronto Chess Tournament

The overflow audience of chess enthusiasts and computer experts attending the International Federation for Information Processing Conference in Toronto watched a team from Northwestern University, Evanston, Illinois, using a Control Data super-scale CYBER 176 computer system, take the world computer chess championship.

Developed over the past seven years by David Slate and Larry Atkin of Northwestern University, the winning computer program, called Chess 4.6, represents 20 to 22 man-years of development time. It is "A" rated by the United States Chess Federation and has a performance rating of high expert. According to David Levy, international chess master who handled the commentary for the Toronto tournament, with continued development, in a few years this class of computer program will be unbeatable by a human player.

Computer experts recognize computer chess as having gone beyond a game and become a critical exercise in developing machine intelligence. Control Data experts explain that chess is a game of infinite variety demanding massive computations from which the computer can come to a conclusion where there is no "yes" or "no" answer.

The CYBER 176 computer system used by the Northwestern team is one of the largest commercially available computers and works at speeds of between 12 and 15 million instructions per second. The system used for the tournament was located at Control Data's engineering laboratories in Arden Hills, Minnesota and passed instructions for each move over a telephone linkage to Toronto. For each move, the computer analysed every possible move for the next six plays and then analysed up to 22 moves beyond likely moves. This meant an average of more than 400,000 positions were examined per move and in some instances up to several million.

(From "INFO", Volume 7, No. 8. Published by Control Data Canada Ltd., Willowdale, Ontario)

The world's computer chess champion, David Slate (part of programming team of Slate and Larry Atkin).

David Slate, an expert-rated chess player, said that the tournament play is a critical test of the computer system and its performance under pressure. "In chess," he said "anything can happen. And does."

There were 16 competitors in the tournament representing Canada, the U.S.A., USSR, Sweden, the Netherlands and the German Federal Republic. The largest computer system competing was the CYBER 176.

Most of the entries were affiliated with university computer research groups. In the case of the winning team from Northwestern University, the computer system time was donated by Control Data and it was Control Data's Dr. David Cahlander who adapted the Chess 4.6 program to the CYBER 176. Dr. Cahlander is a Computer systems expert who worked on the development of the Network Operating System for the CYBER 176 which enables it to handle chess as well as scientific, engineering or business functions in an interactive mode.

Other chess and systems experts at Control Data's development centre involved in the development of Chess 4.6 were John Douglas, an A-rated chess player, Paul Higgins, a systems expert who operated the system throughout the tournament, and another Control Data employee, Chuck Fenner.

At the final match of the four-game, Swiss-style tournament, the Northwestern team met "Belle" the entry of Bell Telephone Labs of New Jersey, which had won twice and drawn once, the CYBER 176 emerged after more than 50 moves undefeated.

The second World Computer Chess Championship was sponsored by the International Federation for Information Processing. The third such event is expected to be held in conjunction with the Federation's next congress scheduled in 1980.

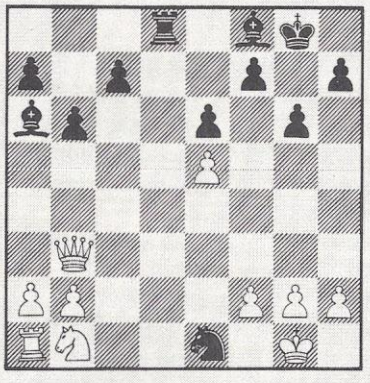
The games and standings

. . . . . At the end of the second round at Toronto the six first-place leaders had been reduced to two, (*CHESS 4.6* and *DUCHESS*) each with two points. Three teams were in 2nd place ties (*BELLE, CHAOS,* and *DARK HORSE*) all with 1½ points. Six teams were jammed up in third place ties (*KAISSA, MASTER, BLACK KNIGHT, ELSA, WITA* and *OSTRICH*) with one point apiece. *CHUTE 1.2, BS '66 '76,* and *BLITZ V* were locked in a 4th place tie with ½ point each. At the bottom of the pile were *BCP* and *TELL.* The eight games of the third round follow

White – CHUTE 1.2	Black – BLITZ V
1. P-K4	P-K4
2. N-KB3	N-KB3
3. NxP	P-Q3
4. N-B3	NxP
5. Q-K2	Q-K2
6. P-Q3	N-KB3
7. QxQch	BxQ
8. N-R3	N-N5
9. P-KR3	BxN
10. PxB	0-0
11. N-N5	N-R3
12. B-K3	P-B3
13. NxRP	N-B4
14. BxN	PxB
15. NxP	PxN
16. QRP-R4	R-K1
17. 0-0-0	B-Q3
18. P-N3	R-K4
19. K-N1	KR-K1
20. K-N2	P-R3
21. R-KN1	R-K8
22. RxR	RxR
23. P-B3	B-R7
24. R-R1	B-K4
25. K-B2	N-Q4
26. P-QB4	N-N5ch
27. K-Q2	R-N8
28. K-K3	RxP
29. K-K4	B-Q5
30. R-R2	R-R6
31. R-N2	R-R8
32. B-K2	R-K8
33. P-B4	RxBch
34. K-B3	R-Q7
35. P-R5	NxQP
36. R-N3	RxP
37. K-K4	N-N7
38. R-QN3	NxP
39. R-N8ch	K-R2
40. P-R6	RxPch
41. KxR	B-K4ch
42. K-B5	BxR
43. K-K4	N-Q3ch
44. K-K3	P-B4
45. P-R4	P-N5
46. PxP	PxP
47. K-K2	P-N5
48. K-K3	P-QB4
49. K-Q5	P-N6
50. K-K3	P-B6
51. K-Q3	P-N7
52. P-R7	BxP
53. KxP	P-N8=Q
54. K-N3	Q-QN8ch
55. K-R4	Q-N7
56. K-R5	Q-R6 mate.
	Black Wins

White – *OSTRICH*	Black – *BLACK KNIGHT*
1. P-K4	P-QB4
2. N-KB3	N-QB3
3. B-B4	N-B3
4. N-B3	NxP
5. BxPch	KxB
6. NxN	P-Q4
7. NxP	P-K4
8. N-N3	B-KB4
9. 0-0	B-K2
10. P-Q3	P-Q5
11. R-K1	Q-B2
12. Q-K2	B-Q3
13. B-Q2	Q-N3
14. P-B3	KR-K1
15. PxP	PxP
White resigns (Machine Problem)	

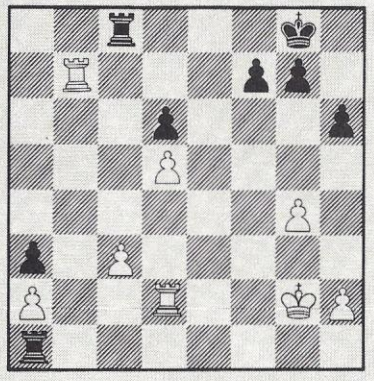
White – *BELLE*	Black – *DARK HORSE*
1. P-K4	N-KB3
2. P-K5	N-Q4
3. P-Q4	P-K3
4. B-Q3	N-QB3
5. N-KB3	B-K2
6. 0-0	0-0
7. P-B3	P-Q3
8. R-K1	PxP
9. PxP	P-QN3
10. Q-B2	P-N3
11. B-KR6	N4-N5
12. PxN	NxNP
13. Q-N3	NxB
14. BxR	BxB
15. R-Q1	B-QR3
16. N-K1	NxN
17. RxQ	RxR
18. Q-R4	P-QN4
19. QxB	N-B7
20. N-B3	NxR
21. QxNP	R-Q7
22. Q-B1	N-B7
23. Q-B1	B-R3



Position after 17th move (Black moves R x R). Early loss of Black's queen makes eventual outcome predictable.

24. N-K4	R-Q4
25. N-B6ch	K-N2
26. QxN	RxP
27. N-K8ch	K-B1
28. Q-B3	R-Q4
29. Q-R8ch	K-K2
30. P-KN3	P-QB3
31. N-B6	R-Q8ch
32. K-N2	B-N4
33. NxP	B-R3
34. Q-QN8	B-N2
35. QxPch	R-Q2
36. Q-B5ch	R-Q3
37. Q-N5ch	K-Q2
38. N-B6ch	BxN
39. QxB	K-K1
40. P-QR4	R-Q4
41. Q-QB3	P-QB4
42. P-R5	R-B4
43. P-R6	P-B3
44. P-R7	R-Q4
45. P-R8=Qch	K-K2
46. Q-N7ch	R-Q2
47. QxBPch	K-K1
48. Q7-B8ch	R-Q1
49. QxP mate	

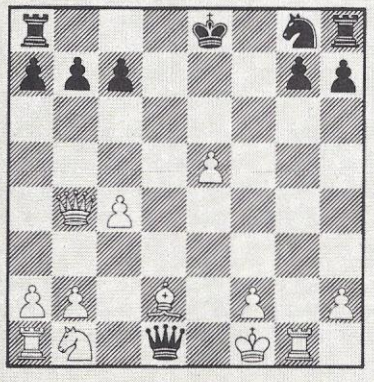
1. P-K4	P-K4	White – *TELL*	Black – *BCP*
2. N-KB3	N-QB3		
3. B-B4	N-KB3		
4. N-B3	B-K2		
5. 0-0	0-0		
6. B-Q5	P-Q3		
7. BxN	PxB		
8. P-Q4	B-N5		
9. PxP	BxN		
10. QxB	PxP		
11. R-Q1	B-Q3		
12. B-K3	P-QR4		
13. B-B5	Q-N1		
14. BxB	PxB		
15. QR-N1	P-R5		
16. P-QN4	R-B1		
17. Q-B5	R-Q1		
18. P-N5	PxP		
19. RxNP	Q-B1		
20. QxQ	KRxQ		
21. R-Q3	R-R3		
22. R-N4	P-R6		
23. R-N7	P-R3		
24. P-B3	R(R3)-B3		
25. N-Q5	NxN		
26. PxN	R-R3		
27. R-Q2	R-R5		
28. P-N4	P-K5		
29. PxP	RxP(K5)		
30. P-B3	R-K8ch		
31. K-N2	R-QR8		
32. R-B2	RxRP		
33. R(N7)xP	R(R7)xP		
34. R(B7)-B3	RxR		
35. RxR	P-N4		
36. R-B6	P-R7		
37. R-N6ch	K-B2		
38. R-N7ch	KxR		
39. P-R4	PxP		
40. K-R3	P-R8=Q		
41. K-N2	Q-Q5		
42. K-R3	QxP(Q)		
43. K-R2	R-B7ch		
44. K-R3	Q-R8 mate		



Position after Black's 31st move. Black is sure to Queen a pawn and win the game.

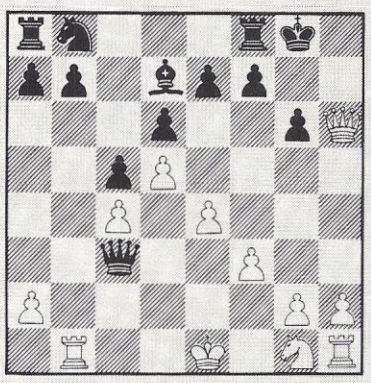
1. P-K4	P-Q3	White – *ELSA*	Black – *MASTER*
2. P-Q4	N-KB3		
3. N-QB3	P-KN3		
4. N-KB3	B-N2		
5. B-K2	0-0		
6. 00	P-QN3		
7. B-KN5	B-QN2		
8. P-K5	N-K5		
9. NxN	BxN		
10. B-Q3	B-QN2		
11. Q-K2	PxP		
12. PxP	P-KB3		
13. B-QB4ch	K-R1		
14. QR-Q1	Q-B1		
15. B-R4	Q-N5		
16. P-KR3	Q-R4		
17. P-KN4	Q-R3		
18. R-Q3	PxP		
19. BxP	QxP		
20. BxR	QxPch		
21. K-R1	QxB		
22. BxBch	KxB		
23. QxPch	K-R3		
24. Q-K3ch	K-N2		
25. Q-K7ch	Q-B2		
26. Q-K5ch	Q-B3		
27. QxPch	Q-B2		
28. Q-K5ch	Q-B3		
29. QxQch	KxQ		
30. KR-Q1	P-KR4		
31. P-B4	P-KN4		
32. P-R4	P-N5		
33. R-Q6ch	K-K2		
34. K-N1	BxN		
35. R(Q1)-Q2	P-R5		
36. P-R5	N-B3		
37. PxP	PxP		
38. R-Q7ch	K-B1		
39. R-Q1	BxR		
40. RxB	N-K4		
41. P-N3	N-B6ch		
42. K-N2	K-K2		
43. R-QN1	R-R7		
44. R-QB1	R-N7		
45. R-B3	K-KB3		
46. R-B1	RxNP		
47. P-B5	PxP		
48. RxP	R-N1		
49. R-KB5ch	K-K3		
(invitation to stalemate)			
50. RxN	PxRch		
51. KxP	R-KN1		
52. K-K4	P-R6		
53. P-B4	R-KB1		
54. K-Q4	P-R7		
55. K-Q3	P-R8=Q		
56. P-B5ch	KxP		
57. K-Q4	R-B8		
58. K-Q3	Q-K1		
59. K-Q5	Q-K4 mate.		

1. P-Q4	P-Q4	White – *WITA*	Black – *BS '66 '76*
2. P-QB4	P-K4		
3. PxKB	P-Q5		
4. P-K4	P-KB4		
5. B-Q3	N-QB3		
6. KN-B3	B-QN5ch		
7. K-K2	PxP(K4)		
8. BxP	B-KN5		
9. Q-QR4	P-Q6ch		
10. K-KB1	Q-Q2		
11. BxN	QxB		
12. QxB	BxN		
13. PxB	QxP(B6)		
14. R-KN1	P-Q7		
15. BxP	Q-Q8ch		
16. B-K1	Q-Q6ch		
17. K-KN2	Q-N3ch		
18. K-KR1	Q-K5ch		
19. KR-N2	P-QR4		
20. Q-QB3	Q-QB4		
21. RxP	0-0-0		
22. QxRP	Q-K5ch		
23. P-KB3	QxPch		
24. K-KN1	Q-K6ch		
25. K-KR1	Q-K5ch		
26. K-KN1	Q-K6ch		
27. K-KR1	Q-KB6ch		
28. KN1	Q-K6ch		
Draw through repetition			



Position after Black's 15th move. White is driven into an endless series of checks.

1. P-Q4	N-KB3	White – *CHAOS*	Black – *KAISSA*
2. P-QB4	P-QB4		
3. P-Q5	P-Q3		
4. N-QB3	P-KN3		
5. P-K4	B-KN2		
6. B-K2	0-0		
7. B-KN5	P-KR3		
8. B-K3	Q-N3		
9. Q-Q2	N-N5		
10. BxN	BxB		
11. P-KB3	B-Q2		
12. BxKRP	BxB		
13. QxB	QxP		
14. R-N1	QxNch		
15. Q-Q2	QxQch		
16. KxQ	B-B1		
17. N-K2	K-N2		
18. P-KR4	R-R1		
19. N-B4	N-Q2		
20. P-R5	P-N4		
21. N-Q3	R-QN1		



Position after Black's 14th move. The exchange of queens leaves Black with the advantage.

22. P-B4	N-B3	24. K-K3	N-KN6
23. P-K5	N-K5ch	25. R-R2	PxKBPch
		26. KxP	NxP
		27. K-B3	B-B4
		28. R-Q1	BxN
		29. RxB	PxP
		30. R-KR3	N-B3
		31. R-N3ch	K-B1
		32. R-K3	P-N4
		33. RxP	PxP
		34. R-B5	P-B6
		35. P-Q6	PxP
		36. K-K3	R-K1ch
		37. K-Q3	P-B5ch
		38. KxP	R-B1ch
		39. K-N3	P-B7
		40. RxN	P-B8 (Q)
		41. KNR-B3	R-N1ch
		42. K-R4	Q-B5ch
		43. K-R5	Q-N4ch
			Black wins

1. P-K4	P-K4	White – *CHESS 4.6*	Black – *BLACK DUCHESS*	34. R(N1)-K1	B-B6
2. N-KB3	N-KB3			35. B-KR6	K-N1
3. P-Q4	PxP			36. R(K1)-K3	B-K5
4. P-K5	N-K5	19. R-K1	P-N3	37. BxB	PxB
5. QxP	N-K5	20. N-B3	P-B4	38. P-KR3	RxP(B4)
6. P-Q6	NxP(Q3)	21. N-Q5	K-R1	39. BxR	RxP
7. B-Q3	N-B3	22. B-KB4	B-QN2	40. R(K3)xP	R-N7
8. Q-KB4	P-KN3	23. N-N5	R-B1	41. R-K8ch	KB2
9. 0-0	B-N2	24. N-B6	R-QB2	42. R(K4)-K7ch	K-B3
10. B-Q2	Q-B3	25. N-K6	R-B2	43. B-K5ch	K-N4
11. QxQ	BxQ	26. NxB	R(B2)xN	44. BxR	N-B5
12. N-B3	0-0	27. NxP	KxN	45. B-B1ch	K-B3
13. N-Q5	BxP	28. BxN	KR-Q1	46. R-K3ch	K-N2
14. QR-N1	B-N2	29. B(Q6)-B4	R-Q4	47. B-N5	N-Q3
15. NxP	R-N1	30. B-KN5	R-KN5	48. RxN	KxB2
16. B-KB4	R-Q1	31. P-KB4	N-R4	49. R-K7ch	K-N1
17. N-QN5	B-B1	32. R-K7ch	K-N1	50. R-Q8 mate	
18. B-N5	R-Q2	33. P-N3	K-B1	White wins	

Seattle tournament in its final round

Twelve teams competed for the title of North American Computer Chess Champion this past October. The four round Swiss style tournament was part of the Association of Computing Machinery's Annual Conference which took place at the Olympic Hotel in Seattle. The four rounds had been scheduled over a three day period, Saturday through Monday with a morning and evening round being played on the first day. **David Levy**, International Chess Master from England served as tournament director. The tournament had been organized by **Monroe Newborn**, McGill University and **Ben Mittman**, Northwestern University. **David Levy** played an exhibition match simultaneously against the computer entries on Sunday afternoon. The rules of play were identical to those of regular human tournament play. When a questionable point arose the tournament director had the authority to make a final decision. Other rules were:

Games were played at a speed of 40 moves per player in the first two hours and then 10 moves every 30 minutes thereafter.

The tournament director had the right to adjudicate a game after four and one half hours of total elapsed time.

If a team encountered technical difficulties (machine failure, communications failure or error, or program failure) during the course of a game, the tournament director could allow them to stop their clock as long as necessary, but not to exceed 30 minutes, in order to restore their system. At the end of at most 30 minutes their clock would have to be started again. The tournament director could grant a team permission to stop their clock at most three times during the course of a game, but the

total time that a team's clock could be stopped was 45 minutes.

There was no manual adjustment of program parameters during the course of a game. In the case of failures, the program parameters were reset to their original settings of possible. Information regarding castling status, en passent status, etc., could be typed in after a failure. If at any time during the course of a game, the computer asked for the time remaining on either his or his opponent's clock, this information was provided. However, the **computer** had to initiate the request for information.

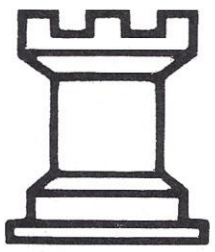
At the end of each game, each team was required to turn in a game listing.

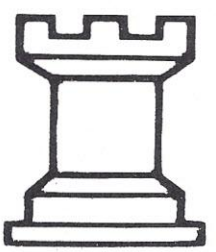
Tie breaking rules for first and second place were established prior to the tournament.

. . . . . After round three, two teams were in first place (*CHESS 4.6* and *DUCHESS*) with 3 points. Four teams were tied for third place with 2 points (*BLITZ V*, *CHAOS*, *XENARBOR* and *8080 CHESS*). In 7th place with one point each were five teams (*BLACK KNIGHT, CHUTE 1.2, OSTRICH, TYRO,* and *WITA*). In last place with no wins to its credit was *BRUTE FORCE*. The six games of the final round follow.

		White – *CHUTE*	Black – *XENARBOR*		
1. P-K4	P-QB3			31. B-N4	P-B3
2. P-QB5	N-QB3			32. B-K6ch	K-R1
3. N-KB3	P-K4			33. B-B5	R-KN1
4. N-B3	N-B3			34. B-N4	R-K1
5. B-K2	P-Q3	18. B-B3	P-KR5	35. B-R5	R-KN1
6. 0-0	P-KN3	19. P-R4	P-R6	36. B-N4	R-KN2
7. P-Q3	B-N2	20. KR-K1	R-QB1	37. B-B8	R-KR2
8. B-N5	N-Q5	21. Q-N7	Q-Q1	38. B-B5	R-R5
9. NxN	BPxN	22. Q-R7	B-B3	39. B-N4	R-R2
10. N-Q5	P-KR3	23. PxP	R-R1	40. B-B5	R-R3
11. NxNch	BxN	24. QxR	QxQ	41. B-N4	R-R5
12. BxB	QxB	25. B-N4	P-KN4	42. B-B5	R-R4
13. Q-R4ch	B-Q2	26. R-B2	Q-Q1	43. B-N4	R-R3
14. Q-R5	P-N3	27. R/2-K2	Q-K2	44. B-K6	R-R2
15. Q-Q5	R-QN1	28. P-B3	Q-B3	45. B-B5	R-N2
16. QR-B1	P-QR4	29. K-N2	Q-B5	46. B-N4	
17. P-QN3	P-R4	30. B-B5	0-0	Drawn by repetition of moves	

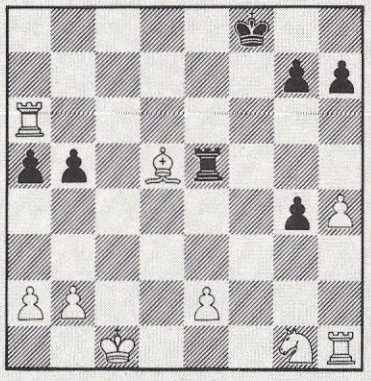
		White – *DUCHESS*	Black – *CHESS 4.6*		
1. P-K4	P-QB4			39. P-N6	P-N3
2. P-Q4	PxP			40. P-B4	K-K2
3. P-QB3	PxP	21. R/R-B1	RxR	41. P-R3	P-B3
4. NxP	N-QB3	22. RxR	R-R5	42. P-R4	K-Q2
5. N-B3	P-Q3	23. B-K3	B-Q1	43. P-N4	P-R3
6. B-QB4	P-K3	24. P-B3	R-R4	44. P-R5	PxP
7. 0-0	N-B3	25. R-Q1	B-KB3	45. PxP	K-K2
8. Q-K2	B-K2	26. R-Q2	R-QB4	46. K-Q4	K-K3
9. R-Q1	P-K4	27. N-K2	R-B3	47. K-B4	K-K2
10. B-K3	N-KN5	28. P-QR4	B-K3	48. P-B5	K-Q1
11. B-Q2	N-Q5	29. N-Q4	R-B8ch	49. K-N4	K-Q2
12. NxN	PxN	30. K-B2	B-Q2	50. K-N5	K-K1
13. N-N5	Q-N3	31. N-N5	BxN	51. K-R4	K-K2
14. B-B4	N-K4	32. PxB	R-QN8	52. K-N4	K-Q1
15. NxP/4	NxB	33. BxP	RxP	53. K-B4	K-K1
16. QxN	B-N5	34. RxR	BxR	54. K-Q4	K-Q2
17. Q-R4ch	B-Q2	35. K-K3	K-B1	55. K-Q5	K-K2
18. Q-N3	0-0	36. B-Q4	BxBch	56. K-Q4	K-K1
19. QxQ	PxQ	37. KxB	K-K1	57. K-B4	K-B2
20. P-QR3	R/B-B1	38. K-Q5	K-Q2	Adjudicated a draw.	





White – *BRUTE FORCE* Black – *8080 CHESS*

1. P-Q4	P-Q4	21. B-Q5ch	K-B1
2. N-QB3	N-QB3	22. RxR	RxP
3. B-B4	N-B3	23. P-K4	P-R5
4. N-N5	P-K4	24. R-QN6	P-N4
5. BxP	B-N5ch	25. PxP	K-N2
6. P-B3	NxB	26. R-N7ch	K-N3
7. PxN	BxPch	27. R/1xP	KxP
8. NxB	N-K5	28. RxP	K-B5
9. NxN	PxN	29. R-QR5	P-R6
10. Q-R4ch	B-Q2	30. R-B7ch	K-K6
11. QxP	P-QN4	31. RxPch	K-Q5
12. P-KN4	0-0	32. R-KN7	K-B4
13. B-N2	P-QB3	33. RxP	K-Q3
14. P-KR4	P-QR4	34. R-R6ch	K-B4
15. 0-0-0	R-R3	35. N-B3	R-K2
16. Q-Q4	P-KB4	36. R-B6ch	K-N5
17. QxB	QxQ	37. P-K5ch	K-N4
18. RxQ	PxP	38. N-Q4ch	K-R4
19. R-Q6	RxP	39. N-N3ch	Resigns
20. BxP	R-B4		



Position after Black's 22nd move: The 8080's early weakness for chasing down and destroying White's minor pieces, plus its delight in premature checks proved its undoing.

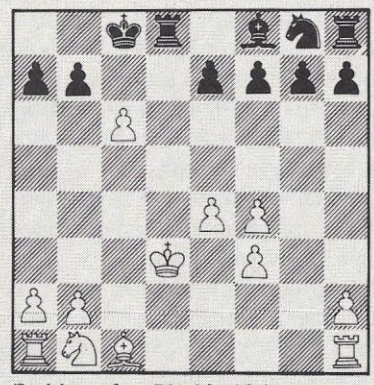
White – *CHAOS* Black – *BLITZ*

1. P-Q4	P-Q4	19. R/R-N1	P-QR4	34. N-B5	R-QB2
2. P-QB4	P-K3	20. B-Q5	N-N5	35. R-B3	B-N3
3. N-QB3	N-KB3	21. BxPch	QxB	36. P-B4	N-B5
4. B-N5	B-K2	22. QxP	B-K3	37. N-R6	R-K2
5. P-K3	0-0	23. P-QR3	QxQ	38. P-K5	B-K5
6. N-B3	P-KR3	24. RxQ	N-Q4	39. P-Q6	R-KB2
7. B-R4	N-K5	25. R/7-QB1	R-QB1	40. N-B7	B-B3
8. BxB	QxB	26. RxRch	BxR	41. P-K6	R-B1
9. PxP	NxN	27. R-QB1	B-N2	42. P-K7	R-B1
10. PxN	PxP	28. P-N3	P-QN4	43. R-Q3	B-Q2
11. Q-N3	R-Q1	29. N-N3	P-R5	44. N-R6	K-B2
12. B-Q3	N-B3	30. N-B5	B-B3	45. K-B2	K-K1
13. 0-0	R-Q3	31. P-K4	N-N3	46. N-B7ch	RxN
14. R/B1-B1	P-QN3	32. P-Q5	B-K1	47. PxR	KxP
15. P-B4	PxP	33. N-N7	R-Q2	48. K-B3	B-B4
16. QxP/4	B-N5			49. R-Q5	B-N5ch
17. N-Q2	R/R1-Q1			50. KxB	N-Q3
18. B-K4	B-Q2			51. RxN	Resigns

White – *WITA* Black – *BLACK KNIGHT*

1. P-K4	P-QB4	21. QxBP	BxP	40. P-N6	P-B4
2. N-KB3	N-QB3	22. BxR	BxB	41. Q-K3	K-N3
3. P-Q4	PxP	23. QxRP	B-B3	42. P-R4	N-K5
4. NxP	N-B3	24. QR-K1	B-B6	43. K-N1	R-N8ch
5. N-QB3	P-Q3	25. P-R3	Q-N3ch	44. Q-K1	RxQch
6. P-KR3	P-QR3	26. K-R2	N-B3	45. K-R2	R-QN8
7. P-KN4	NxN	27. Q-Q1	BxR	46. B-B1	RxB
8. QxN	P-K4	28. RxB	R-B1	47. P-N7	R-QN8
9. Q-Q3	B-K3	29. R-K2	R-B6	48. P-N8=Q	RxQ
10. B-N2	B-K2	30. Q-Q2	Q-R4	49. K-N1	N-B6
11. B-K3	R-QB1	31. R-K3	RxNP	50. P-R5ch	KxP
12. Q-Q2	P-KR4	32. QxQ	RxR	51. K-R2	P-N4
13. P-N5	N-Q2	33. P-QR4	R-K7	52. K-N3	P-N5
14. N-Q5	BxN	34. QxRP	RxP	53. K-N2	P-B5
15. PxB	R-B5	35. Q-R8ch	K-R2	54. K-B2	R-N8
16. Q-K2	P-N4	36. PxP	R-N7	55. K-N2	N-K5
17. P-N3	R-KR5	37. Q-B6	N-K5	56. K-R2	K-R5
18. P-KB4	0-0	38. Q-K8	N-N4	57. K-N2	P-B6ch
19. 0-0	PxP			58. K-R2	P-N6 mate
20. B-B2	P-B6				

1. P-Q4	P-Q4	White – *TYRO*	Black – *OSTRICH*
2. P-QB4	PxP		
3. P-K4	N-QB3		
4. P-Q5	N-K4		
5. P-KB4	B-KN5		
6. N-KB3	BxN		
7. PxB	N-Q6ch		
8. BxN	PxB		
9. QxP	P-QB3		
10. K-K2	Q-Q2		
11. PxP	QxQch		
12. KxQ	0-0-0ch		
13. K-K2	PxP		
14. R-Q1	R-Q3		
15. B-K3	R-R3		
16. BxP	RxPch		
17. B-B2	P-R4		
18. P-N3	R-R3		



Position after Black's 12th move: Black places a rook on an open file from where it proceeds to torment White's defenses.

19. P-R4	R-B3
20. K-K3	R-R6
21. P-N4	P-K4
22. PxP	R/3xPch
23. K-K2	BxP
24. B-N6	N-K2
25. R-Q4	R-K6ch
26. K-B2	B-K8ch
27. K-B1	R/R-B6ch
28. K-N1	R-K7
29. R-Q8ch	K-N2
30. B-Q4	RxP
31. R-Q7ch	K-B1
32. RxN	R-N5ch
33. K-R2	RxB
34. K-N2	R-B7ch
35. K-N3	R-Q6ch
36. K-R4	R-B4 mate

Computer-Chess-Program-Thesis: Part 3

Mike Valenti's thesis on Writing A Computer Chess Program has previously discussed "background" and "language use." Before the actual development of the chess program, the thesis emphasizes the requirement of possessing some chess skill:

Part 3

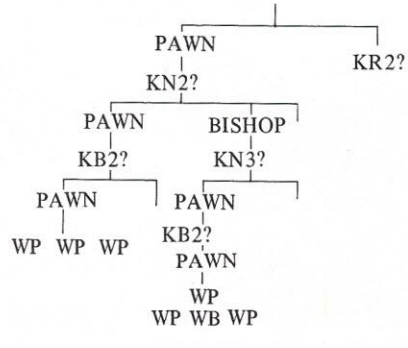
"This section describes some recent discoveries and theories based on studies of chess players (see [Simon and Chase 1973], [Wason 1972], [Michie 1972] and [Baylor et al. 1966]). In particular, some insights have been gained into human perceptual and memory processes, and some work has been done toward human simulation of these [Simon and Chase 1973]. Following this, is a discussion of how these ideas apply or can apply to the chess program itself.

"The exponential explosion of possibilities in exhaustively searching all possible moves to a reasonable depth of four to six plays is far beyond a human player's capabilities, as well as a computer's. Humans seldom search more than about a hundred paths in their "search tree," while considering typically no more than two to three alternatives from each position.

"It was found from studying weak as well as strong players, that there was little difference in the number of moves explored, depth of search and width of search. The only difference was that grandmasters typically chose stronger moves at each position. The question is how were they able to do this? These players were given reasonable chess positions to study briefly (2 to 10 seconds) and then told to reproduce the board from memory. Grandmasters and masters were able to reproduce almost perfectly (around 93%) positions with about 25 pieces. However, players classified as expert did significantly worse (around 72%) and the drop-off was quite sharp. Good amateurs faired about 50% while novices did about 33%. When the same players were given random boards to reproduce, all players did equally poorly (about 15 to 20%). Thus the masters exhibited no better visual imagery than weak players. Their perceptual skills were chess-specific.

"Studies were then made with regard to the eye movements of a chess expert as he first looked at a position (first 5 seconds). The human eye moves abruptly from one fixation to the next at the rate of 4 to 5 per second. In this time there were about 20 fixations centered around the pieces that a chess player would consider important, with few fixations around the edges of the board or on empty squares. These fixations would move from one piece to another studying pieces in chess-specific relations to each other, such as a knight attacking a pawn, or a bishop pinning a king. This evidence was inconsistent with the hypothesis that the subject was searching through a tree of plausible moves in the first 5 seconds.

"While perceiving this information, the expert would memorize these "chunks" of chess-specific information and relate them to known chunks in long-term memory. These chunks of information could be individual pieces, relating colour, size and shape with familiar objects. These pieces then became parts of new chunks of more complex objects. The tree below is a representation of this process. The terminal nodes (which are the new units or chunks), represent 3 pawns in a row on the second rank and a fianchettoed bishop. (For example, the symbol KR2? means what piece is on king's rook two square? WP means white pawn and WB means white bishop.)



"Another concept is required in this development, and that is the concept of **short-term memory**. Short-term memory [Simon and Chase 1973] is postulated to be a memory system with a constant capacity of around 7 chunks. This memory is used to temporarily hold the information before recalling it in the board reproduction experiment. The data stored in memory are the names of the chunks. Furthermore, if a player can recognize larger groups or more groups of pieces, he will then be able to reproduce a more accurate board.

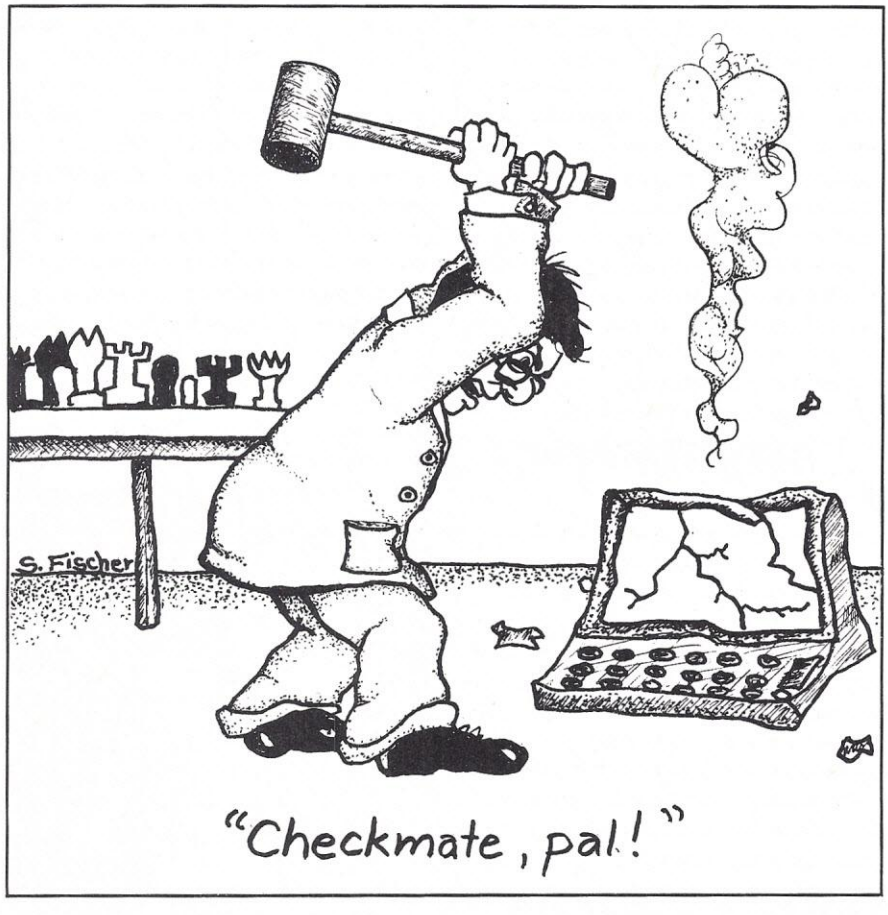
"Experiments [Simon and Chase 1973] bear out these theories to some extent. The two researchers were able to estimate the size and number of chunks that masters and experts used. The average size of the first chunks recalled by masters and experts were 3.8 and 2.6 respectively, and the average number of chunks were 7.7 and 5.7, respectively. However, the theory does not explain differences in the number of chunks, and the model, therefore, seems to be a bit too simplistic.

"Simon and Chase claim that masters have an edge over weaker players because of a capability to recognize a large number of chunks on sight, crudely estimated to be 50,000 or more (comparable to a highly literate person's vocabulary of words). This vocabulary is built up following thousands of hours of familiarizing chess positions – which explains why there are no instant masters in chess.

"These postulates on human perception and memorization are termed a 'perceiver-memorizer theory.' This program does not resemble this 'perceiver-memorizer theory,' but it does have a capacity for recognizing chunks of information. A direct simulation of these postulates is probably not necessary at this time in order to produce a good chess-playing program, but these studies certainly suggest some possible strategies. This program was begun after these ideas were published, but they are relevant in considering future possibilities using this work as a basis.

"My program typically examines a few hundred paths in looking ahead compared to a human player's hundred or so. However, its vocabulary of chunks is very limited at this time, and any advantage it may have is that it can consider more possibilities plus the fact that it never overlooks anything that it **can** recognize. Heuristics, that are applied to each move, check for things such as two pawns side-by-side or three pawns on the same diagonal on adjacent rows or pieces pinning other pieces. Most recent improvements in the program have been made in the form of making these heuristics recognize more chunks. However, there is also a lot of room for improvement in the look-ahead procedure."

(Part 4, next month, will begin the actual design and implementation of a chess program itself.)



Response from SD

. . . . . In the San Jose Microcomputer Chess Tourney SD CHESS finished in a tie for 6th place out of a possible 11 positions. The question was raised on why *Ira Baxter's* SD CHESS could not beat the other entrants, considering that SD CHESS has 32K of memory – more than any other entrant. BORIS has 2½K ROM and ¼K RAM and yet managed to earn a tie for 2nd place. Responding to this query, *Ira* says, of his SD CHESS: "I agree. One would think that 32K of memory should beat (an average of) 4K. It just goes to show that an IBM 360 with a megabyte of memory does not necessarily outperform a 4-function calculator for the intended purpose, given the proper application.

"First – what went into that 32K? Here's the list: 1) The SD Runtime Package – 10K bytes of stuff that knows how to execute compiled BASIC programs. Of this, 4K or more is devoted to doing floating point arithmetic – which SD CHESS does not use at all. 2) Graphic display data. SD CHESS at the San Jose tournament drew a picture of its own chess board for all to see (this is **not** part of the standard SD CHESS program) on a home-brew raster scan graphics terminal. Most of the display data is comprised of 24x24 bit arrays of piece pictures – for a memory space consumption of 3.4K, not counting the code to manipulate the graphics display. 3) The SD I/O package, a simple system that allows compiled BASIC programs to read and

write files to a disk, print on Hytype (under software control) move data streams to and from another computer, etc. Essentially this is a primitive operating system. Memory requirements: 4K bytes. 4) The chess program itself – in compiled BASIC 14K bytes. Total: 29K (I wasted the other 3K).

"What I'm mostly paying for is generality – my micro does a lot more things than play chess, whereas COMPU-CHESS can do nothing else. The point is, if you make the purpose of an object simple enough – it will be simple compared to the general-purpose object (re the calculator vs the IBM 360). Very nearly half of my memory was used for things that understood nothing about chess – and therefore contributed nothing to the performance of the program.

"Now, why did SD CHESS get beaten at San Jose? First and foremost – time. Even in spite of being compiled, compiled BASIC programs are still ten times **slower** than cleverly coded assembly language. (The reason people buy BASIC compilers is that the resulting programs run ten times **faster** than **interpreted** BASIC programs.) If one examines the games of SD CHESS vs the CHALLENGER and BORIS, he will see that SD CHESS lost on **time** – 120 minutes of CPU compared to 5 or 10 minutes for the CHALLENGER or BORIS. If SD CHESS were ten times as fast it would have had a total of 12 minutes clock time – very nearly even. One also sees that the positions are more or less equal when SD CHESS ran out of time. It would be extremely interesting to see what would have happened had the games been played **without** a clock. Moral: if you want to run a program fast, code it cleverly in machine code.

"I did it in BASIC partly for fun and mostly because it only took me two weeks of evenings to build the program. I'm sure the people that built SARGON (the winning team) invested much more effort in building their program. Against the two BASIC programs (*Watson* and *Tenberg*) SD CHESS did very well. These two programs played positionally only. *Watson* went home much chagrined convinced that his program absolutely needed lookahead to prevent it from doing downright stupid things. (He finished in last place.) For instance, – in one game, *Watson's* program posted a knight where it forked two major pieces and a pawn. What better position can one ask for than that? Well. . . how about the same position in which the knight is susceptible to capture for free? *Watson's* program lost that knight. . . and the game.

"*Tenberg* played a very good positional game. (He, too, was tied for 6th place.) It was so interested in position, in fact, that it moved its white bishop clean through two of its own pawns to mount a vicious attack. This kind of bug should not have showed up at the tournament. In spite of the illegal attack (we couldn't get *TENBERG* BASIC to take the move back and decided to continue for the heck of it,). SD CHESS managed to hold onto its marbles and the attack petered out about the same time as the tournament did.

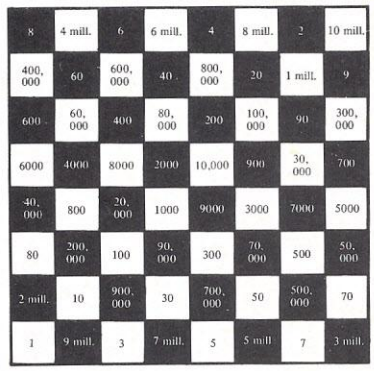
"Against *Watson* and *Tenberg*, we decided essentially to ignore the clocks (all the programs were BASIC and slow). Only the game between SD CHESS and *Watson* was completed to checkmate. Here, SD CHESS rammed a set of pieces down on *Watson's* castled king, stripped off his pawn cover, and queened a pawn to force checkmate. I was extremely surprised at SD CHESS's end game – it was much better than I thought against *Watson.*

"I will admit that SD CHESS needs better positional play. *Mark Watson* and I had plenty of time to discuss the differences between our two programs while they played each other. A note, now, concerning your chart in last month's issue. My impression was that TENBERG BASIC played on an LSI-11. (I've never seen an F8 with diskette drives and a nice BASIC. Perhaps *Tenberg* will send in an explanation or correction.)

"I was very impressed with BORIS and the other chess-playing commercially available 'boxes'. They are relatively easy to use, and play well enough to give a rank amateur like me a hard time. In conclusion, I want to point out that the San Jose awarded three 1st places: 1) Machine code by individuals with 8K; 2) Commercial boxes (CHESS CHALLENGER, BORIS, etc.) and 3) BASIC programs. SD CHESS took first place in category 3."

Checkerboard Calculator

. . . . . **John R. Allen**, of Portsmouth, N.H. writes: "I happened to see the enclosed picture in an encyclopedia in the library. It is supposed to be a checkerboard used as an ancient calculator. The encyclopedia said it was used in Asia and worked like an abacus. Stones were piled up on the squares where needed. But the book didn't say how the thing worked. It merely



traced the English word 'Exchequer' to this board. I've tried to figure out how they could use stones and turn the checkerboard into a calculating machine. But I give up. Can't do it. Would some of your readers please send in an explanation."

Midwest Exposition

. . . . . The second annual Midwest PERSONAL COMPUTING Exposition, sponsored by PERSONAL COMPUTING magazine, will be held at the Chicago Expocenter, October 5, 6, 7 and 8, 1978 (Thur., Fri., Sat., Sun.). As matters now stand, the world champion Chess 4.6 computer program with its authors David Slate and Larry Atkin are scheduled to conduct a seminar in computer chess. There is even some activity in the direction of perhaps having David Levy come over (from London), at that time, and publicly defend his wager that he would beat any computer chess program before the end of 1978. Organized this year by Industrial and Scientific Conference Management, Inc., the show will feature 200 exhibitors displaying the

latest computer equipment, peripherals, accessories and software by America's leading manufacturers. An expanded seminar program, besides the possible chess seminar, will teach new developments, innovations and practical applications in the personal computing field. For further announcements and developments concerning the show, keep an eye on the RANDOM ACCESS department in this magazine.

One-armed player

. . . . . From **Timothy M. Bonham**, D605/1630 S 6th St., Minneapolis, MN 55454: "You may have heard of the success of *CHESS 4.6* running on the new Control Data Cyber 176. This program was entered in the Minnesota open chess tournament and made it into the finals, eventually finishing in 6th place (out of 6). The program did better in the eliminations than in the final. In the elimination in its first match it beat an "A" rated player who was expected to be a strong contender. This is apparently the first time any computer chess program has beaten an A player in an open tournament and will be entered in the Guiness book of records. This victory seemed to scare many of the human players there. Then, in play against Minnesota's only chess Master (who wasn't playing in the tournament) the program beat him in three games of speed chess. This scared many players even more: even though he went home and studied some and came back to beat the program in two games the next day. Some students at Northwestern have obtained a surplus mechanical arm from NASA and are building a microprocessor controller to operate the arm and emulate a terminal with an operator inputting opponents' moves to the main computer. I would like to see more on this project.

Computer Chess Movie

. . . . . In Holland Mr. B. Swets (of the ICCA) is planning a film on computer chess in English. The film will be designed for showing during conferences and meetings of information processors as well as scientists and psychologists. Expected to be a 25 minute, 16 mm. documentary, the film, which will be suitable for TV broadcasting, will introduce the public to the subject and will promote the current interest in computer chess. Electronic soundtrack will be provided by Utrech Institute of Sonology. Most of the movie can be shot in the Netherlands but to make it truly international Mr. Swets' organization is seeking cooperation from an American filmcrew and producer with a sound financial backing.

North Star Micro-Disk System

The North Star MICRO-DISK SYSTEM is a complete mini-floppy disk system for 8080 and Z-80 microcomputer systems which utilize the Altair bus structure.

Included in a North Star MICRO-DISK SYSTEM are:

one North Star controller board

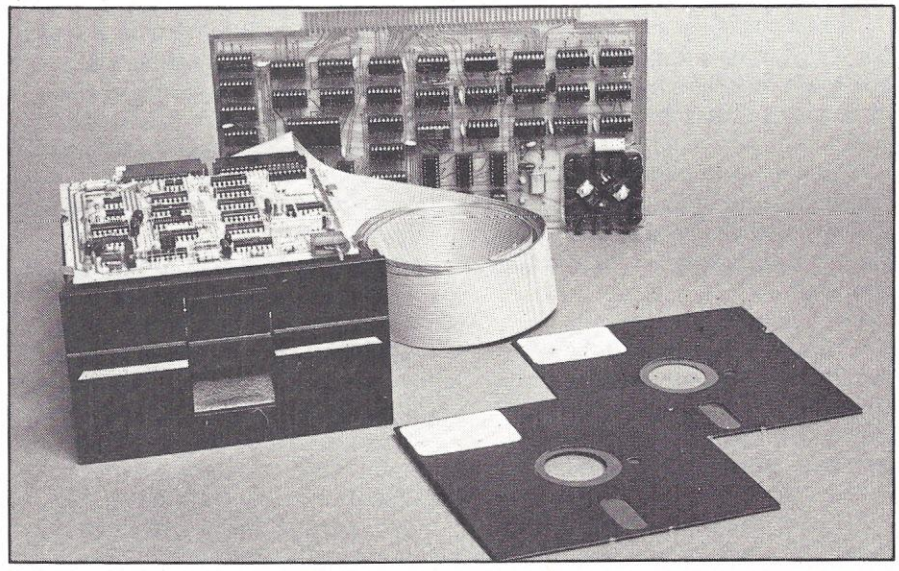
one Shugart SA-400 minifloppy disk drive

one power supply card (to supply power to the disk drive)

cabling and connectors between controller and drive

two mini diskettes – one with DOS and N.S. BASIC

hardware and software documentation



Hardware

The North Star system is complete – however, one must read the component description carefully. The Shugart SA-400 drive included with the system comes complete but bare. That is, the drive comes without a cabinet – something most hobbyists will want. The cabinet is available though – at additional cost. You should also note the power supply board, attached to the disk drive, requires raw power from the host mainframe. Prospective customers should make sure their mainframes have a large enough power supply to handle the added load. Once again, a stand alone power supply is available from North Star at additional cost.

The North Star disk controller is a single Altair bus compatible PC board that interfaces the computer system to the disk drive. The controller circuitry is made up of Schottky and low power Schottky integrated circuits, and has its own crystal-controlled clock on-board.

Also on-board is the system bootstrap in PROM. This PROM is memory mapped to take up 1K of system RAM space starting at E800 hex in the standard version. (The bootstrap start address is E900 hex.) Non-standard versions are available upon request also at additional cost.

Since the system is not interrupt driven, you must be careful that an interrupt does not occur during disk operations. Such an interrupt would cause the system to miss data travelling to or from the disk at the time of the interrupt. You can avoid such problems by disabling the interrupts in the system before initiating a disk operation.

The disk drive is the Shugart SA-400 minifloppy, probably the most widely used of the small-sized floppy disk drives on the market today. The floppy diskettes used are hard-sectored for 256 bytes per record, 10 sectors per track, and formatted by the system for 35 tracks to make possible storage of up to 89.6K bytes.

Software

The North Star system arrives with two standard pieces of software: the disk operating system, or DOS; and North Star disk extended BASIC. The DOS occupies 2.5K bytes of system RAM memory starting at 2000 hex. This 2.5K includes 256 bytes of space for user I/O routines. For the uninitiated, this means that there is space provided in the DOS for I/O routines written especially for the system in which it is running.

For example, if one had a Poly-88, the input and output routines would surely be different from those for an IMSAI 8080. Having that extra space reserved for personalized I/O routines means that the user has to attach his (or her) own I/O drivers only once – then store them away onto a system diskette which when bootstrapped into the system, will then automatically set up the I/O for the system.

The North Star DOS manual has the source listing for the I/O area of their code, and an example listing for a personalized DOS, as well as instructions for going about the personalization process.

This system requires a minimum of 16K RAM starting at 2000 hex in order to run DOS and BASIC. DOS alone may be run in as little as 8K of RAM memory.

A disk operating system is a set of commands that may be sent to the computer in order to get various pieces of information onto or off of disk storage. The commands available in North Star DOS are two-letter mnemonics followed by any required arguments. The commands available are as follows:

LI <optional drive #>

This command will list the contents of the diskette directory.

CR <filename> <length> <optional start address>

This command will create a new file on the diskette. The length indicates how many 256 byte blocks are reserved. The start address is for where on the diskette the file will be. If no start ad-

dress is given, the new file will appear after the previous file on that diskette.

DE <filename> <filetype>

This command will delete a file from the directory.

CO <optional drive #>

This command is used to compact the file space on a diskette. Any unused disk space between files will be freed by moving all files toward track 0.

TY <filename> <filetype>

This command is used to set or change the type of the specified file. Type 0 = default, 1 = machine language file, 2 = BASIC program file, 3 = BASIC data file.

GO <filename>

This command will load the specified file into memory and start execution. The GO command can be used only with machine language files that have been assigned a "go-address".

GA <filename>

The GA command sets the "go-address" for a machine language file. The go-address must be the first byte of the program.

JP <hex address>

This command is used to make the processor jump to the specified address and start execution.

LF <filename> <hex address>

SF <filename> <hex address>

These commands are used to load or save a file to or from RAM. The entire amount of information that will be transferred is equal to the number of blocks reserved for the file.

CF <source filename> <destination filename>

This command may be used to copy one file to another. The destination file must be at least as large as the source file.

CD <source drive #> <destination drive #>

This command will copy an entire diskette from one drive to another. Obviously, this command only applies to multiple drive systems.

RD <disk address> <hex address> <# of blocks>

WR <disk address> <hex address> <# of blocks>

These commands can be used to read or write information from diskette to RAM or from RAM to diskette. The disk address is a block number between 4 and 350.

IN <optional drive #>

This command must be used to initialize a diskette before using it for the first time.

DT <optional drive #>

This command may be used to test a drive or verify the usability of a diskette. A tested diskette must be initialized again before use. This program is an "endless" one. Operation will be stopped by a hard disk error or by a control-C typed on the console device.

North Star Extended BASIC was designed for use on any 8080 or Z-80 microcomputer system, and is available in three versions. Version 5-FPB is a non-disk extended BASIC designed for use with another of North Star's products, the Hardware Floating Point Board. Version 6 is North Star's Disk Extended BASIC which comes standard with every disk system. Finally, version 6-FPB is the Disk Extended BASIC, also for use in conjunction with the Hardware Floating Point Board. Version 6 is available with arithmetic precision from 2 to 14 digits – 8 digits of precision is standard.

Version 6 has the commands to store and load programs to and from disk as well as the ability for BASIC programs to perform both sequential and random access disk file processing. By using the CHAIN statement, one program can call another into memory and automatically start execution. All temporary variables are destroyed, so if variables must be passed from one program to the next, they must be stored in a scratch file before chaining to the next program.

General Review

North Star does not release any of the source listings for their software except what is needed to attach the I/O drivers to. This makes the system a take-it-or-leave-it system to some extent, but except for the most die-hard operating system software tinkerer this will probably be acceptable. This is, however, the only real "bad mark" we can give them. Anything else depends more or less on the particular likes or dislikes of the user.

The North Star MICRO-DISK SYSTEM is an excellent accessory. We found it easy to plug into every system we tried: Altair, IMSAI, PolyMorphic Processor Technology, Vector Graphic, TDL and several home-brew systems employing the Altair bus structure. All the system's software resides in memory at the same time (DOS, BASIC, and user program, or DOS and user program in machine language) making it a very usable system, even with only one disk drive.

We call this system a manual disk operating system. That is, the operator must take care of all the "housekeeping" for the system.

As an example, consider saving a BASIC language program that has just been perfected: Step 1 would be determining how many blocks of 256 bytes have been used. You must know how many free bytes are in the system when there's no program in memory. To determine how many blocks of diskette space will be needed, find out how many free bytes there are with the new program now in memory, and subtract this number from the original number of free bytes. Divide by 256 and round to the next greater whole number to find the number of 256 byte blocks needed to store the program.

For Step 2, leave the BASIC interpreter to go to the DOS. Do this by typing BYE with a carriage return. Once in DOS, use the CR command to create the directory listing: CR TEST xx, where TEST is the program name and xx is the number of blocks needed to store the program.

Step 3 is to "type" the file by using the TY command. Type: TY TEST 2, where once again TEST is the program name, and 2 indicates that the file is to hold a BASIC program.

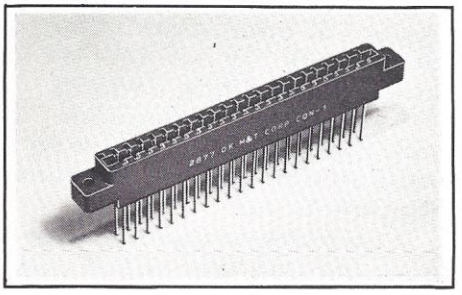
For Step 4, re-enter BASIC with a warm start so as not to destroy the existing BASIC program. You can do this with the JP command to jump to the warm start address of BASIC: JP 2A00.

Now back in BASIC, having done all the "housekeeping" to allow us to save the program, type SAVE TEST to save the newly created BASIC program onto diskette in the file space.

If you want to add disk storage to your existing system, at a reasonable price, and if you're looking for a proven operating system with a good disk-extended BASIC, then this might be the system for you. □

Stephen Pereira owns his own North Star Disk System and is the President of Microcomputers, Inc., Nashua, NH.

Systems, Subsystems, Software



The OK Machine and Tool Corporation of New York has a new **Edge Connector** (Model CON-1) that features wire wrapping contacts. The contacts are nickel silver over beryllium copper and feature bifurcated/bellows design that provides constant pressure while minimizing contact distortion and stress. Connector body is molded of U.L. and MIL approved VALOX (an insulating material with claimed superior dielectric, thermal and chemical characteristics). CON-1 is priced at $3.49. *Circle No. 102*

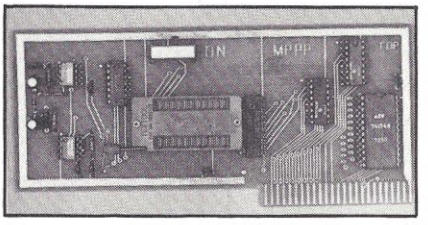
Two new **4096-bit Schottky PROMs** have been added by Texas Instruments to its bipolar memory products. The PROMs, designated as SN54S/74S476 and SN54S/74S477, are organized as 1024 words of 4 bits. The 74S commercial version features 35 ns typical access time, 30 ns maximum enable time and 60 ns maximum access time. The 54S military circuit has the same typical access time, 40 ns maximum enable time and 75 ns maximum access time. Other features are dual select lines to simplifiy memory expansion and permit fast, easy programming, the company says. Supplied in the industry-standard pin configuration, these new PROMs can serve as direct replacements for virtually any currently offered 18-pin 4K PROM, according to Texas Instruments. Prices range from $9.30 to $18.60 depending on the various models ordered and in quantities of 100. *Circle No. 103*

Some **Micro-Business software** applications have been introduced by Arkansas Systems of Little Rock. The business-application software currently runs on an 8080/Z-80 system with a CP/M Operating System. The software is written in FORTRAN and uses an Indexed Sequential Access Method; therefore, says the company, it will run quite fast. Presently released are the General Ledger and Payroll systems. The General Ledger is designed for CPAs and is said to be generalized and flexible. There are more than 20 programs in this system alone, including Profit and Loss Statements, Balance Sheets, verification of valid accounts, automatic entry of I/E totals in balance sheet and other features that, says the company, one would expect to find only on a large computer.

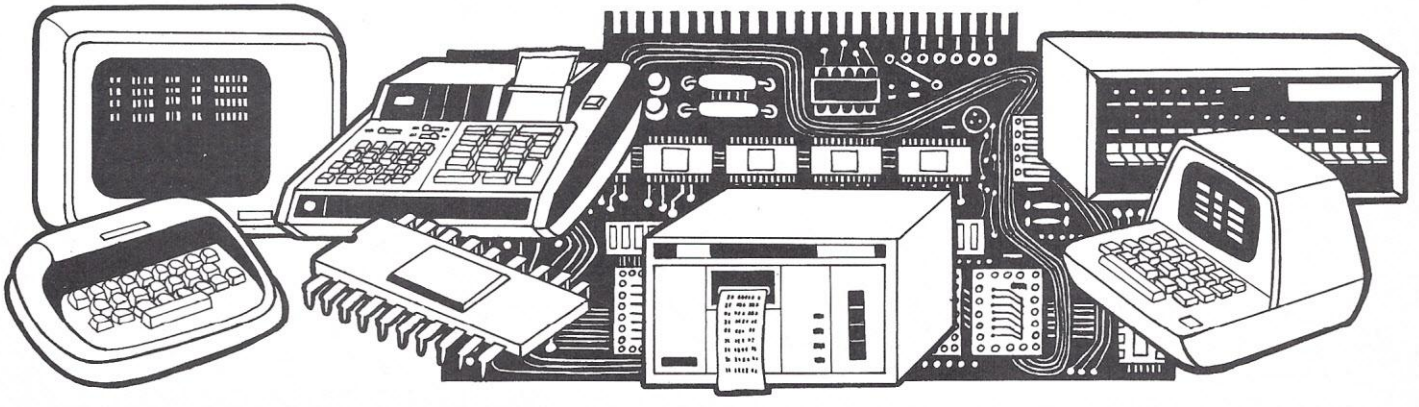
The second program currently available is the Payroll system which prints payroll checks, calculates multi-state taxes, local taxes, multi-pay periods, bonuses, W2s, department reporting and other features. Cost of the General Ledger or Payroll is $775 each. Under development, and scheduled to be released shortly, are Accounts Payable and Accounts Receivable, each at $495 (estimated price). *Circle No. 101*

Panasonic's Electronic Components Division has a new 40-pin DIL plastic **4-bit microcomputer.** Designated the MN1405 Model, the device offers on-chip functions of ALU, 2048 x 8 ROM, clock generator, two 4-bit input ports, one 4-bit and 8-bit output port, two sense lines, one 12-bit discrete output port and 128 x 4 RAM with four words that are directly accessible. The company claims this device is ideally suited for applications such as electronic games, appliance controllers and a variety of random logic applications. *Circle No. 108*

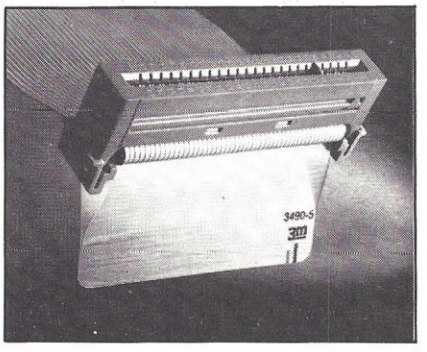
Microproducts of Hermosa Beach, CA, has a new **Apple II EPROM Programmer.** The new device consists of a fully assembled, double-sided, fiberglass printed circuit board with plated through holes and gold-plated edge connector. It plugs directly into any



available slot on the Apple II board. A Textool zero insertion force socket permits delicate handling of the EPROM. The new EPROM programmer is fully contained and has a 25-volt power supply for programming the Intel 2716 EPROM. Price of the EPROM programmer is $89.95. The Intel 2716 socket adapter is $9.95. *Circle No. 105*



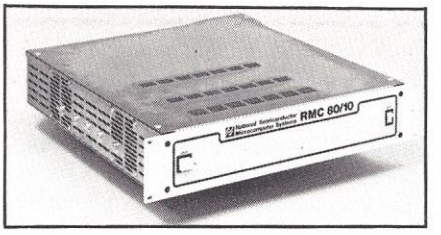
The 3M Company, of St. Paul, MN, has a new 50-position **Scotchflex card-edge connector.** The company claims that this connector is more reliable and rugged than many of its predecessors. Designated the 3415-0002 Connector, the device incorporates flanges into which the strain relief clip can be firmly snapped. An optional pull tab allows quick disconnection and reconnection with minimal flexion and wear. The strain relief and option tab permit flat



cable to be guided straight out from the back of connectors. PC boards can therefore be stacked tighter, improving density of packaging. *Circle No. 110*

A new **alpha-numeric display and keyboard interface** device is available from Matrox. Called Model MTX-B1, the alpha-chip keyboard portion provides all scanning signals to debounce and decode any keyboard of up to 64 keys. The display portion of the MTX-B1 provides the timing and refresh signals needed to drive any number of characters from 1 to 32. The chip has a 32 x 8 display refresh RAM built in as well as an ASCII character generator for both 7 and 14 segment displays. Timing and refresh signals are created internally for both the display and the keyboard functions. In addition, the MTX-B1 includes an intelligent controller and bus interface. The chip can be interfaced to any TTL, CMOS or NMOS microprocessor through an I/O port or bus. Price of the device is $49 single or $39 in 100s lots. *Circle No. 111*

A new **multi-level power supply,** from National Semiconductor of Santa Clara, CA, has been designed especially for Series/80 microprocessor-boards, although the BLC 635 Power Supply may also be used for other microprocessor systems. Main features of the new multi-level power supply include its ability to combine line and load reg-



ulation with current limiting, with overvoltage protection and with power-failure detection. The power-supply device provides +12V at 2.0A, +5V at 14.0A, -5V at 0.9A and -12V at 0.8A. Incorporated are circuits to limit current at 1.2 times rated values at all levels and overvoltage circuits which trip at 1.16 to 1.32 rated voltage. Load regulation is 0.1 percent for a 50% load change and line regulation is 0.1% for a 10% line variation. The BLC 635 with cables for the Series/80 products is priced at $460. *Circle No. 112*

Rair Black Box, an integrated **microcomputer** from Britain, incorporates floppy disk drives. The unit's dual disk drives accept 5 1/4" single sided, reversible, soft sectored floppy disks, each with a capacity of 81,920 bytes. Each

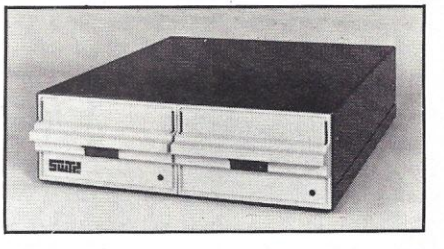
disk has 40 tracks, each track 16 sectors and each sector 128 bytes. Transfer rate is 125,000 bits/sec; latency, 100 milliseconds; and access time, 40 milliseconds track to track.

The microprocessor handles up to 80 instructions. It has 256 input/ouptut channels. Dynamic MOS RAM contains up to 65,536 bytes. Software includes a disk operating system, an extended BASIC interpreter, relocatable FORTRAN compiler and a COBOL compiler. Other features include dual serial input/output ports, an eight-slot motherboard for system expansion and a transparent ROM bootstrap. *Circle No. 126*

Stevens-Arnold, of South Boston, MA, has a new "UB" Series **DC/DC Converter Supply** for microprocessors. The unit delivers 7 watts of low noise isolated-regulated power with outputs of +12V at 500mA and -5V at 200mA. There is a ±0.05% regulation and an average voltage tempco of ±0.01%°C. Standard DC inputs accepted include 5V, 5.8V, 12V, 15V, 24V, 28V, 48V or 60V. Rated efficiency is typically 55% with an operating temperature range from -25°C to 71°C with no derating and with free-air convection cooling. *Circle No. 113*

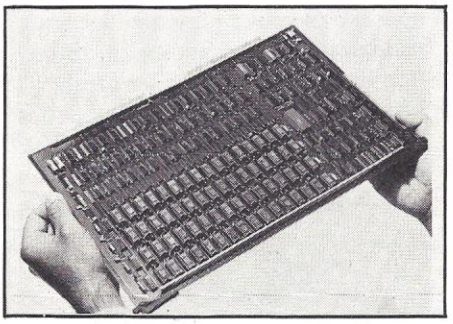
A low-cost **digital-to-analog converter,** the DAC336-8 series, has been introduced by Hybrid Systems of Bedford, MA. A compact device, ready to use, and claimed to be adjustment-free, each DAC336-8 includes precision reference, ladder network, switches, output amplifier and input storage register. The hybrid IC design is factory pre-trimmed to ±0.05%. Simple pin jumpering allows the user the choice of four voltage range outputs and requires only 200mW of power. The unit accepts TTL, DTL and 5V CMOS logic levels at a minimum delivery output of +5mA at ±10V. Price of the unit is $24 for the less expensive of two models. *Circle No. 124*

A dual drive, single density, double sided 8" **floppy disk system,** Model DMAF1, is available from Southwest Technical Products of San Antonio, TX. The hardware includes an SS-50 bus (SWTPC 6800) and a compatible DMA (direct memory access) controller capable of handling up to four drives. The company claims that its supplied software is one of the most powerful microcomputer disk operating systems yet available for either the hobbyist or businessman. An 8K BASIC interpreter with disk file capability and string functions is also included with the system. Each diskette holds approx-



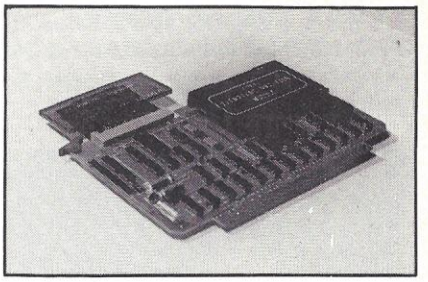
imately 600,000 bytes of data and, with two drives, more than one megabyte of data on line is achieved. Price of the system is $2095 assembled or, in kit form, $2,000. *Circle No. 114*

A **refresh memory system** designed for image-enhancement applications has been announced by Intel Corp. of Sunnyvale, CA. The new system, designated in-5770, is mounted on one 11.25" by 16" edge connector-type printed-circuit board and utilizes Intel 16-pin, 16K MOS dynamic



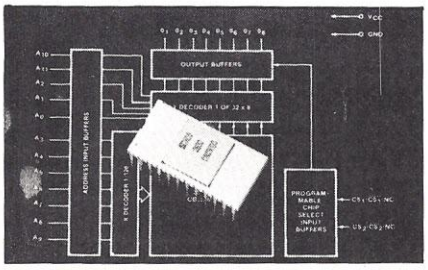
RAMs. It has a total capacity of 256K four-bit words organized into four image planes of 256K one-bit words per plane. In addition to image enhancement, the in-5770 refresh memory allows a user to engage in image processing. For example, says Intel, with the original 512 x 512 image stored in one or more memory planes, the user could load segments of the image into another memory. This feature would permit zooming in and enlarging certain specified areas of the image for closer inspection. Various modes of operation are possible, such as parallel read, single-bit read, parallel write, single-bit write, parallel read-modify-write, single-bit read-modify-write, serial read and clear mode write. Price for the 512 x 512 board is $3,400. A 256 X 512 board is available for $2,400. *Circle No. 104*

Summa/11 computer, from the Interpring Group, is suitable for business, scientific and personal applications, according to the company. The computer uses the DEC 16-bit LSI-11 processor and includes the PDP-11 instruction set. Other features include 8 general purpose registers, 8 addressing modes, vectored priority interrupts, power-fail/auto-restart, 256K-byte floppy disk, 24K RAM serial or parallel terminal interface. Software includes Omni/OS Disk Operating System and OmniBASIC. *Circle No. 125*



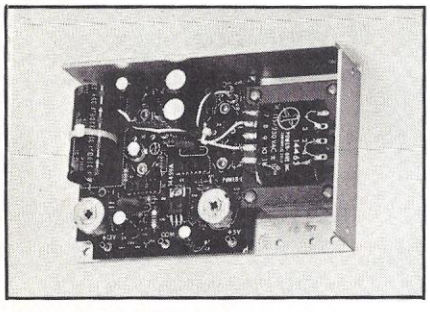
A troubleshooting and repair assist is available from Phoenix Digital's LS-100 Series **Digital Signature Analysis products.** Verification of correct digital patterns provides GO/NO-GO testing as well as diagnostics, 99.99% error detection accuracy, rapid identification of bad components, PC boards and entire systems. Options offered include remote LED signature display, 32-line multiplexer, logic probes, enhanced software package and stand-alone test ability. Applications include home computers, industrial and process control, product characterization and production testing. Prices begin at $295. *Circle No. 115*

A **32K MOS mask ROM** that is fully static and in an industry-standard 24-pin package is a new product from Signetics of Sunnyvale, CA. The unit,



Model 2632, is TTL-compatible, operates from a single +5-volt supply and features maximum access time of 450 nanoseconds. The new ROM is designed for memory applications where high performance, large bit storage and simple interfacing are important design objectives, says Signetics. The unit has two programmable chip select inputs as well as OR-tie compatibility on the outputs to facilitate memory expansion. Price for the 2632 MOS ROM is $11.50 each when ordered in minimum quantities of 500 per pattern. *Circle No. 106*

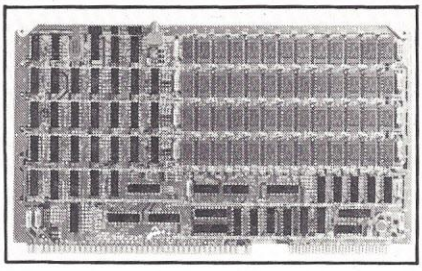
A **disk-drive power supply** dedicated to the 5.25" minifloppy has been announced by Power-One of Camarillo, CA. The new model, designated CP-249, provides dual outputs of 5V at 0.7A and 12V at 1.1A. Standard features include 115/230 VAC ± 10% AC input capabilities, ± 0.05% line and load regulation and full protection against short circuit and overload. Maximum ripple is 3.0 mv PK-PK, while transient response is



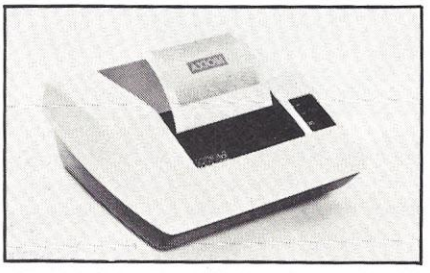
30 μ for a 50% load change. The new unit, says the company, is ideal for the 5.25" floppy since it eliminates the need for special power supply design or custom units. Single quantity pricing is $39.95. *Circle No. 107*

A versatile **plotter,** Model DFT, is available from Sylvanhills Labs, Pittsburgh, KS. Some possible applications of the plotter include engineering drawing, event recording, printed circuit artwork, computer generated art, X-Y transport for small drills, tape verification for numerical control, computer graphics and games. Designed for the hobbyist, businessman, engineer or teacher, the plotter can hold pencils, felt-tip pens, engineering pens, LED lamps, scribe points or any tool up to 3/8" diameter. Plotting business trends, maps, and schematics are among its uses. It can also drill PC boards and position a 3" square table and a chemical flask weighing two pounds under different spigots for automatic chemical analysis. The plotters require 8-bit parallel I/O ports and 5 and 24 volt power sources. A basic 8080 software program is included in the owner's manual. Prices range from $795 to $1,300 depending on size ordered. All come in kit form. *Circle No. 121*

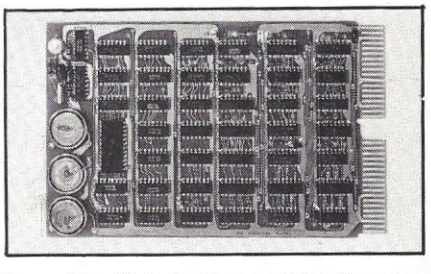
Mupro, of Sunnyvale, CA, has introduced a line of **Compatible Memory Boards.** These boards, says the company, fill the need for higher data reliability and integrity together with easier maintenance in SBC-80 systems by providing error-correcting logic. Seven memory sizes are offered: 4K, 8K, 12K, 16K, 32K, 48K and 64K bytes. The smaller boards are available with 4K dynamic RAM; larger boards, with 16K RAM. Three configurations are available for all eight sizes: without error detection; with single-bit parity; or with single-bit error correction and double-bit error detection. All error-correcting configurations are equipped with diagnostic indicators to pinpoint the troublesome memory chip. Prices begin at $455. *Circle No. 116*



A **compact printer,** the EX-820, which mixes high resolution graphics and full ASCII alphanumerics, is available from Axiom, Glendale, CA. Driven by an Intel 8048, the EX-820 contains one of the most sophisticated microprocessor-based controllers available on any printer, regardless of size or cost — according to the claims of an Axiom spokesman. The printer uses 5" wide electrosensitive paper. The size of the graphic field can be chosen from four pre-programmed horizontal dot resolutions up to 128 dots per inch. Vertical dot resolution is fixed at 65 dots per inch and there is a provision for automatic histogram generation. In addition to exceptional graphics capabilities, says the company, the EX-820 is a versatile line printer in its own right. Hardcopy is produced by electrosensitive printing. This is accomplished by passing a high current for a few microseconds from 8 tungsten wires in the printhead to the conducting surface of the aluminized electrosensitive paper. The aluminum is removed almost instantaneously, exposing a layer of black ink which is seen as a dot matrix forming the printout. The EX-820 Printer is priced at $795. *Circle No. 120*



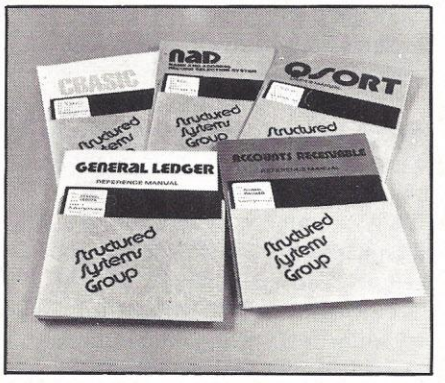
Digital Pathways of Palo Alto, CA, has a dual-size **peripheral board** which provides calendar and real-time functions for DEC's LSI-11 and LSI-11/2. The TCU-50D unit is equipped with rechargeable battery back-up capability. The device will keep track of the current date and time, even when the computer's power is turned off, for up to



3 months. This feature enables the user to maintain records on system downtime during power failures. Units are shipped working and preset to the correct date and local time at the customer's location. A simple software routine is available for resetting the date and time. Price for the TCU-50D is $295. *Circle No. 123*

Structured Systems Group's Business Systems Series of **business programs** is designed to run on 8080 or Z-80 CPUs and CP/M operating systems. Programs available include general ledger, designed for professional accountants and small businesses and priced at $995. The Name and Address System maintains files and allows selection on all fields for printing labels, reports or new files. Price is $79. QSORT is a full disk sort/merge for $95. CBASIC compiler/interpreter includes full disk access, PRINT USING and 14 digits of precision for $99.95. *Circle No. 130.*

Accounts receivable can be processed on your 8080- or Z-80-based CP/M microcomputer system using Structured Systems Group's General Ledger compatible **Accounts Receivable system.** The software package is a full open item billing system for small businesses, designed for business-oriented professionals who need reliable and simple operations, the company says.



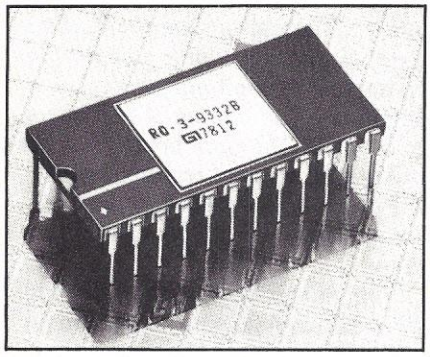
Priced at $750, the A/R system features itemized statements, trial balance reports, late charges, reminder letters, recurring receivables and sales reports. *Circle No. 131*

A low-profile **8-bit microcomputer** from National Semiconductor requires three and one-half inches panel space in a standard 19-inch rack, according to the company. Based on the BLC80/10 central processor board using the INS8080A microprocessor, the rack-mounted computer model RMC 80/10 incorporates programmable serial and parallel input/output, complete busing, power supply, fans and three expansion-board slots.

The unit has six general-purpose 8-bit registers, an accumulator, a 16-bit program counter and a 16-bit stack pointer. The registers may be used singly or in pairs for double precision operations. The 16-bit program counter allows direct addressing of 64K bytes of memory. The stack pointer permits storage and recall of register contents, anywhere in random-access memory, while servicing interrupts and subroutines.

A programmable communications interface provides either synchronous or asynchronous serial input/output with jumper-selectable rates from 110 baud to 19,200 bits per second, either as 20mA current-loop or RS232C signals. System software selects mode and inserts data, control characters and parity bits. Price is $1345. *Circle No. 129.*

A new **32K ROM**, the RO-3-9332, is available from General Instruments. Featuring fully static operation, the ROM is directly compatible with all popular microprocessor families. Fully static operation allows users of 16K ROMs such at Intel's 2316E to directly



replace two 16K ROMs with a single 32K. Two versions of the new device are available: the "B" version with 450 nanosecond access time and the "C" version with 350 nanosecond access. Price for the new ROM is $12 when ordered in large quantities. *Circle No. 117*

Suitable for industrial applications, **Wintek 4K BASIC** is a 6800 interpreter featuring control of in-terrupts, direct memory read and write, assembly language subroutines and flexible input/output. Oriented toward process control and monitoring, this BASIC version may reside in RAM or in PROM for instant power-on operation. When stored in PROM, the interpreter can immediately enter the RUN mode, allowing unattended operation in dedicated applications, developers say.

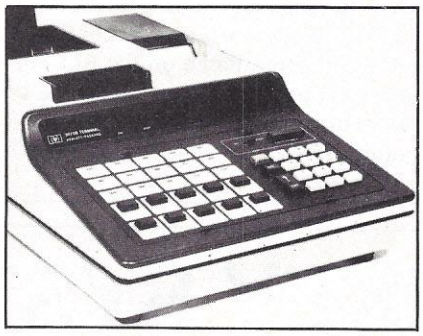
Wintek's 4K BASIC is available on cassette ($95) or in PROM on a WINCE ROM module ($299). The source listing costs $95. *Circle No. 153*

A compact computer terminal, 3070B from Hewlett-Packard, was designed for use in manufacturing operations. The unit offers facilities to collect information and enter it into an HP 1000 Computer System in an orderly way; the terminal can also act as the central controller and data-collection center for several interconnected programmable instruments.

Capable of reading standard type-3 Hollerith badges and marked or punched cards, the terminal has a keyboard for local numeric entries, a 15-character LED display and a quiet 50-lpm, 20-character-wide alphanumeric strip printer. The unit measures 6 by 11 by 16 inches and weighs 13 pounds.

The 3070B offers 15 prompting lights and 10 program-definable keys. In use, the prompting lights might be programmed, on an associated HP 1000 Computer System, into a prompted sequence such as INSERT BADGE, then ENTER WORK ORDER, then ENTER CUSTOMER NUMBER and so forth. Typically, the terminal would deal with such activities as work in progress, production scheduling, cost accounting and shipping and receiving.

Program-definable keys could be preprogrammed to provide YES or NO answers, to start previously-defined subroutines, or to report actions by sending

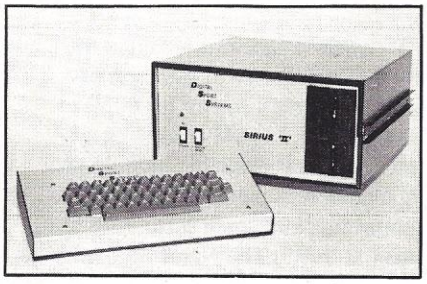


previously agreed-upon sequences of sybmols, H-P says.

Price is $3200 including the multifunction reader for badges and marked or punched cards and the strip printer. *Circle No. 152*

Comptronics offers two F-8 products both for serious hobbyists and design engineers. The first is an **F-8 CPU board** compatible with the S-100 bus. With 3850 CPU and 3853 SMI, the unit provides sockets for 2K of EPROM monitor, two PIO sockets, and connections for six I/O ports. The board has 64 bytes of scratch pad RAM and a fully buffered data bus. The unit sells for $239 as a kit or $275 assembled.

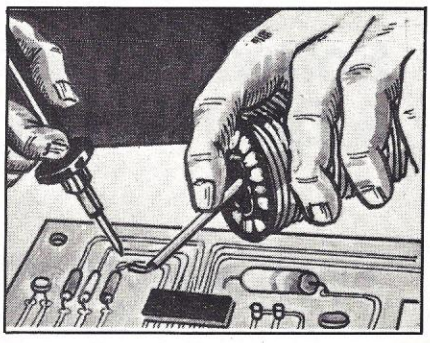
The second product is an **F-8 microcomputer** with keyboard and six-digit display. The unit provides audio interface and speaker compatibility with the on-board KD-BUG (3856) music routine, 2K of RAM expandable through an S-100 connector, and 1K of EPROM with 4 additional 2708 sockets. Prices are $375 for kit and $425 assembled. *Circle No. 154*



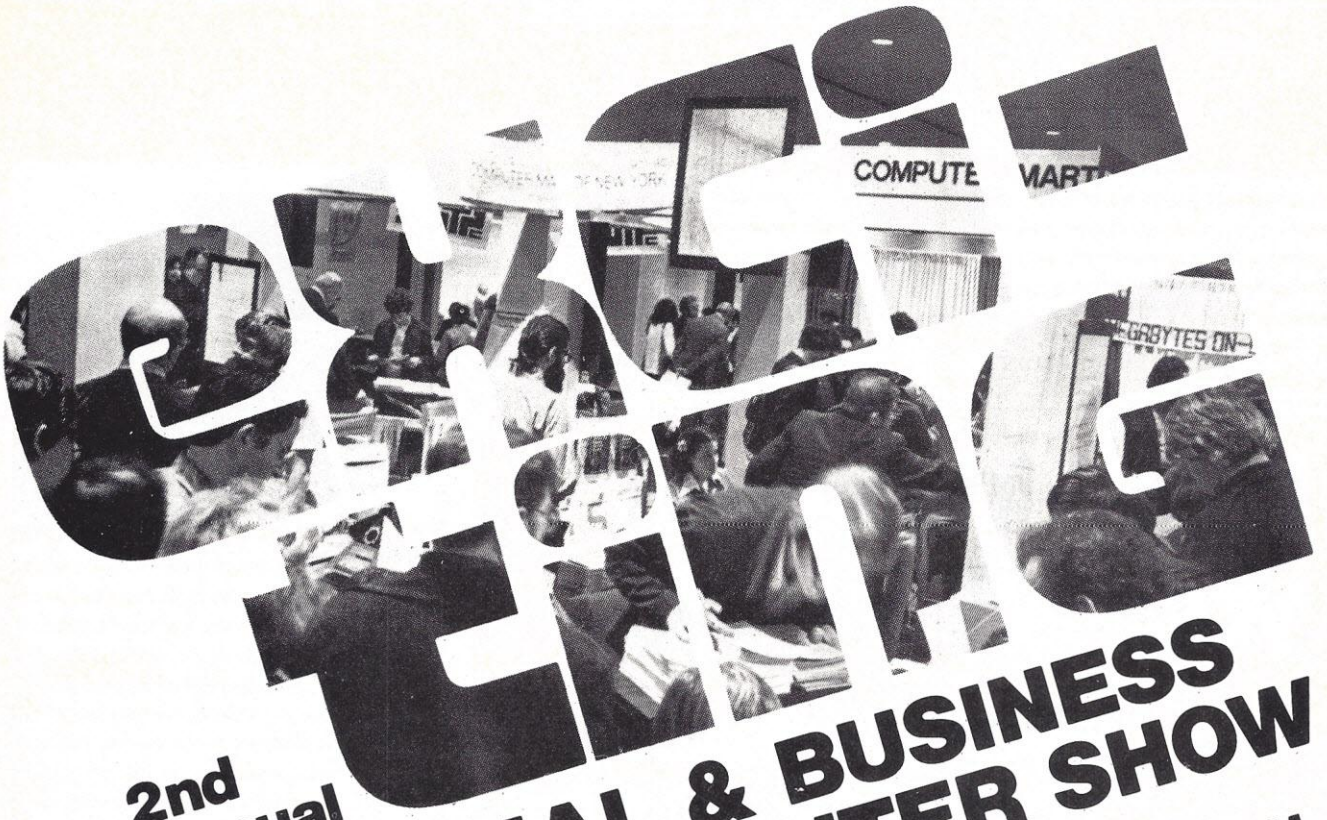
Sirius II Home Computer from Digital Sport Systems uses two microprocessors, the Mostek Z-80 (for main computations), and a Fairchild 3870 that handles keyboard and TV interface overhead. Also included are 32,768 characters of read/write memory, RS-232 interface for I/O, 8192-character PROM board with keyboard with alphanumeric and graphic capabilities and TV interface. Other features included are a full disk operating BASIC interpreter, monitor and software programs (home management, personal finance, educational programs, process control and games). Sirius II can also run business software.

Available hardware includes additional RS-232 I/O, additional parallel I/O, add-on 16, 384 RAM cards, A/D-D/A converters and additional mini-floppy disk drives. Price is $1850. *Circle No. 127*

Chemtronics of Hauppauge, NY, has a **new soldering routine** which it calls the SD5 Modular Solder/Desolder System. The system has a desolder wick dispenser tool that snaps into the core of a one-pound spool. The SD5, says Chemtronics, offers a simple, practical solution to the problems of service people, technicians and production-line workers who must often alternate between soldering and desoldering. The retractable desoldering tool has a 2-1/2"



heat-resistant Teflon probe, permitting pinpoint wick application. The probe also facilitates "webbing", or shaping of the wick for greater solder absorbtion. *Circle No. 109*



2nd Annual

PERSONAL & BUSINESS SMALL COMPUTER SHOW

NEW YORK COLISEUM
September 15-17, 1978

EXCITING

That's the word that sums up the dynamic small computer show in New York that was established last year at the Coliseum. The lecture areas were filled by interested people, the exhibits were great, the attendance was the highest of any small computer show in the country.

The big point is that everyone who came went away happy. Accountants, hobbyists, lawyers, doctors, brokers, retailers, business people, programmers, research scientists and just plain homeowners.

The second big point is that we're doing it again. This time, bigger and better. Top seminars, top exhibits. Small computers from $500 up, software and kits.

Save time. Get your tickets in advance by mail.

NAME _____
(please print)

ADDRESS _____

City _____ State _____ Zip _____

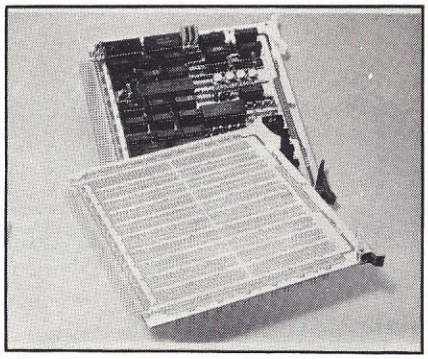
Send me tickets in advance for Personal & Business Small Computer Show, Sept. 15-17, 1978, in the New York Coliseum.

(Check One) One Day ($5)__ Two Days ($9)__ Three Days ($13)__

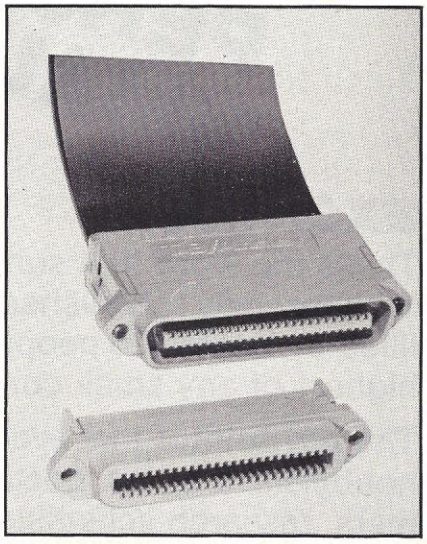
Send check or money order (U.S dollars) payable to Personal & Business Small Computer Show to 78 E. 56th St., New York, N.Y. 10022.

PERSONAL & BUSINESS SMALL COMPUTER SHOW
78 East 56th Street, New York, N.Y. 10022

Hybricon of Littleton, MA, announces a high density microprocessor **"Wire Wrap" board** that is bus compatible with the Zilog Z-80 Microcomputer. The 2-Z-80 Board features 36 columns of 59 plated thru holes on a 0.100" x 0.100" pattern. This arrangement permits mounting any combination of IC DIPs from 8 to 40 pins. An additional feature of the board is a double row of 47 contact holes on a 0.100" spacing at the top of the board for mounting flat cable connectors. The panel contains a combination of 6 power and ground planes which affords the user the advantage of multiple voltage sources while isolating analog and digital grounds. Unit pricing for the board is $68.50. *Circle No. 118*

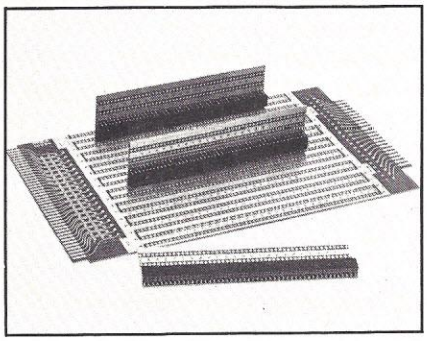


Just introduced to the market is T&B/Ansley's **Ribbon Connector.** The device is designed to IEEE Standard 488 that can be mass terminated, in seconds, to standard 50 mil pitch flat cable without wire stripping or soldering. Designed for I/O applications, the new one piece male and female ribbon connector offers maximum installation versatility through its special connector design. This design features cable insulation, displacing contacts on .050" centers and I/O mating ribbon contacts on industry standard 0.085" centers. The one piece Ribbon Connector offers ease of installation and features self-aligning cable grooves that automatically position each cable conductor over the beryllium copper contacts to assure maximum electrical reliability. Connector prices are as low as six cents per contact. *Circle No. 122*



WHAT'S COMING UP

A **receptacle connector** designed for board-to-board interconnect applications is a new product from Stanford Engineering, Santa Clara, CA. The MODCON Interconnect uses box form contacts that mate with conventional



0.025 x 0.025 wiring posts on 0.100" center-to-center spacing. The device uses spring-tempered phosphor bronze contacts featuring dual cantilever beams housed in a high temperature, glass-filled Ryton insulator. *Circle No. 119*

Columbia Data Products' microprocessor controlled solid state **data buffer**, capable of storing 16,000 char-

acters, was designed to interconnnect between asynchronous RS-232 or TTY compatible terminals and a modem/CPU for store and forward applications. Dual UARTs provide on-line baud rate conversion with speeds selectable from 110 to 19,200 baud. Features include search and edit routines, auto answer, Transparency Mode, self-contained diagnostics and FOX message generation to aid in system debugging. Price for the data buffer is $995. *Circle No. 128*

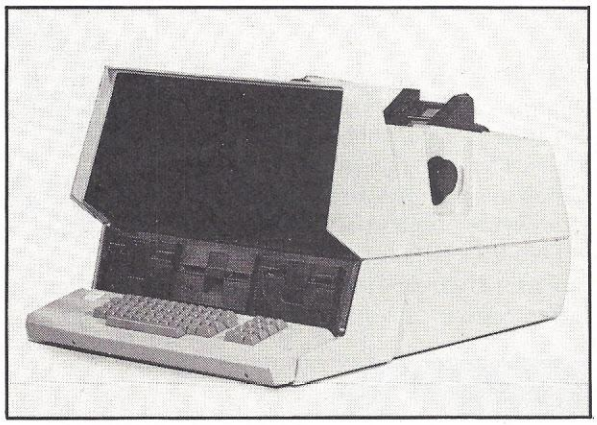
WHAT'S COMING UP

General Robotics Corporation announces a self-contained **desktop computer system,** the MVT/X3, which includes a DEC LSI-11, 62K bytes of RAM, line printer, keyboard, three mini-floppy disks, operating system, and 12 line x 40 character display.

The central processing unit is a DEC LSI-11 packaged in an eight quad slot backplane. Floating point arithmetic hardware (EIS/FIS) is a standard feature. The normal configuration has nine additional dual height QBUS slots for user expansion. MOS RAM provides 62K bytes of self-refreshing memory with an access time of 450 nanoseconds.

The line printer prints an original plus 4 copies at 60 characters per second in 5 x 7 dot matrix format, and has forms tractors for paper input and output. A 64-character ASCII set is standard.

Three mini-floppy disks comprise the on-line storage, with total disk storage exceeding one megabyte.



The flat plasma display panel features 12 lines of 40 characters and neon-orange characters.

System software options include FORTRAN IV, BASIC and APL. The hardware may be expanded with external floppy disks or cartridge disks. Typical applications are small business systems, data stations, network nodes, and laboratory and special purpose installations, the company says. Price is $12,000. *Circle No. 136*

Quay Corporation, has reduced the price of its Q80AI, Z-80-based, **S-100 compatible microcomputer** from $550 to $350. The factory assembled and tested unit includes 1K static RAM, 1K PROM resident monitor, on-board PROM programmer, keyboard interface and serial (RS232C/TTY) I/O. Quay has also package priced the Q80AI, the Q80SMB (8K static memory board) and the Q-TBPE-80 (Palo Alto Tiny BASIC extended) at $495. *Circle No. 142*

GRAPH#1 and GRAPH#2 are **utility packages** for using the SWTPC GT-61 graphics display with a 6800 computer.

GRAPH#1 is for use with a machine language program while GRAPH#2 is for use through SWTPC 8K BASIC version 2. When using GRAPH#2, the user can program the GT-61 display directly from a BASIC program without any machine language programming.

Both versions can display and erase points, lines

and the complete upper-case ASCII character set. Additional capabilities include user-defined characters (such as chess symbols or space ships).

Applications for these packages include display of statistical data plus legends. Game possibilities are numerous, since real-time moving displays can be created, even from BASIC, say the developers, Applied Microcomputer Systems. Each package costs $5.50 (for paper tape), or $6.50 (Kansas City Standard Cassette) postage paid. Package includes a 22 page instruction manual with sample programs and source listing. *Circle No. 143*

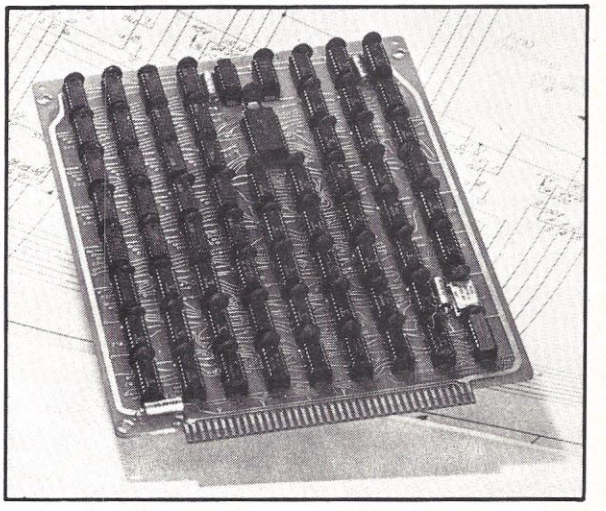
R. F. Applications, Inc., announces its model 2002 digital **mobile telecommunications system.** This system produces hard copy output in vehicles via existing two-way radio equipment. Applications include law enforcement and commercial fleet vehicle use, according to the company. The microprocessor-controlled base terminal features recall, editing and selective calling, and can store and display up to 45 words per transmission. Transmission rate is 180 words per minute. Each mobile receiver costs $995 and the base terminal costs $2490; leasing arrangements are also available. *Circle No. 149*

Applied Data Communications' Event 2000, a series of **business computer systems** with dual double density floppy disk drives, gives users 1.25 Mbytes on-line; systems will accommodate up to 6 additional floppies for a total of over 5 million bytes on-line. The 2000 can also accommodate up to four 10-megabyte disk cartridge units. Features include 48K RAM memory expandable to 64K, peripheral interface capabilities, and programmable communications. The 2000 also employs Applied Data Communications' version of Business BASIC – the MicroDOS/BASIC Operating System. Applications packages include General Ledger, Accounts Receivable, Accounts Payable and Inventory Control. The basic Event 2000 is priced at $11,300 and includes microprocessor-based computer with 48K memory, 1K PROM bootstrap loader, dual double density floppy disk drives, 60 cps keyboard teleprinter and workstation desk. *Circle No. 133*

Mini-Dynamo, a **simulation package** for minicomputers, is now available from Pugh-Roberts Associates. A large-computer version of this continuous simulation language has been used for dynamic modeling of industrial, social and engineering systems. The mini version supports Dynamo simulation on PDP-11, NOVA, Eclipse and Varian minicomputers. Mini-Dynamo offers standard Dynamo equation formats, tabular and graphical output, automatic sorting of equations for correct computational order and rerun of model without recompiling. The one-time license fee is $2500 ($1000 for educational institutions). *Circle No. 134*

Mailing Label and Directory Lookup Package, compatible with North Star disk operating systems, is available from HSC Computer Services. The package, written in North Star BASIC, allows entry of up to 480 records on each diskette, listing of all records on file, lookup of name in file with retrieval of address and telephone number, and printing of mailing labels. Category mailing is permitted with use of attribute code on each record. Price of the package on diskette with documentation is $50. *Circle No. 132*

T-1000, from Dynamic Sciences, is a **minicomputer emulator** that can be microprogrammed to run the instruction sets of mini or microcomputers. Bipolar construction permits 300,000 operations per second. Byte-slice architecture allows emulation of computers with 8, 12, 16 or 32-bit words. The CPU contains 16



full word (16-bit) hardware registers. Up to 65,536 words of memory can be directly addressed with 262,000 words of extended addressing available.

This binary computer derives its emulation control from a variable length microprogrammable memory. Although the basic computer is contained on a single printed-circuit board, additional memory, programmed I/O, various interfaces and a series of data option modules, such as floating point and memory management, are available on additional boards. Microinstruction execution rate is 10 MHz. A direct memory access (DMA) channel is also provided. The price for the T-1000 minicomputer emulator, depending on the computer being emulated, starts at $1000. *Circle No. 135*

Microsoft now offers **COBOL-80** for 8080/Z-80/8085 microprocessor systems. COBOL-80 conforms to the 1974 ANSI standard. All Level 1 features and some Level 2 options for nucleus and for sequential, relative and indexed file handling facilities are included. Level 1 features include table handling and library and inter-program communication facilities. Level 2 features include the verbs STRING, UNSTRING, COMPUTE, SEARCH, and PERFORM (varying/until),

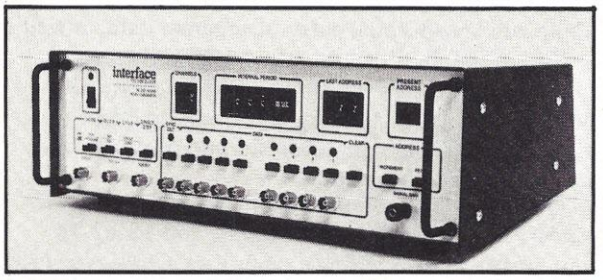
along with condition specifications by condition-names, compound conditions and abbreviated conditions.

The software system consists of two complete packages: a compiler for translating source code into relocatable object code, and a runtime system containing standard routines needed by the object code at execution time. The whole system runs in less than 32K bytes. Rate of compilation is 250 lines per minute. Complete documentation is supplied with the system or may be purchased separately for $20.

COBOL-80 is available to run under the CP/M and ISIS-II operating systems for $750 per copy. *Circle No. 139*

Speakeasy Software offers four **Apple II software packages.** "Bulls and Bears" is a stockmarket game, "Warlords" a medieval battle game, and "Microtrivia" a "game best played at a party, along with plenty of good food and drink", according to the company. (Typical "Microtrivia questions: What happened on Napoleon's wedding night? In *Star Wars,* who was the leader of the Storm Troopers? What do Jumbo and Trigger have in common?). TRS-80 and PET versions on these three games are now under development, the company says. The fourth package is "Kidstuff", an assortment of puzzles, games, riddles, jokes, brain teasers and mazes for grade-schoolers. Each software package retails for about $12. *Circle No. 146*

Interface Technology sells a new self-contained multichannel **word generator** suitable for most general purpose word generator and test applications, the company says. Model RS-600 Word Generator offers from one to eight channel operation with 10 MHz data rates up to 512 total bits. It features separate external clock

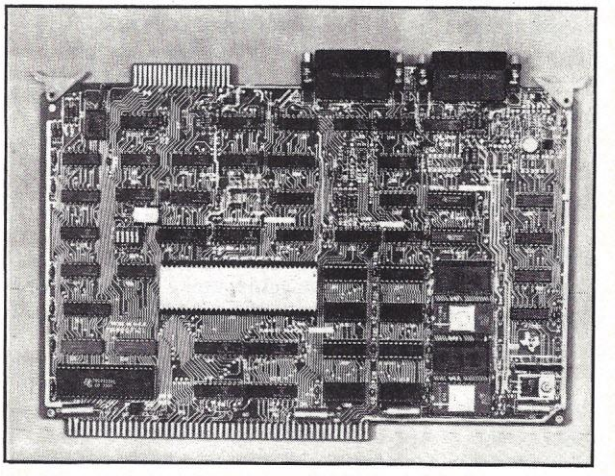


and trigger inputs, separate sync and data clock output, has TTL-compatible inputs and outputs and offers front panel programming.

The unit can be used to stimulate circuitry under test, to stimulate other instruments in gated or triggered modes, to control experiments involving other circuitry and instruments and to emulate digital equipment within a system which isn't available but is required for test purposes, according to the company.

Features include switch selectable output modes ranging from one channel x 512 bits to two channel x 256 bits/channel, four channels x 128 bits/channel or eight channels x 64 bits/channel. Users define the total number of bits to be output. Price is $1,295. *Circle No. 137*

The TM 990/101M, a **16-bit microcomputer module** from Texas Instruments, offers up to 4K words by 16 bits of EPROM memory and up to 2K words by 16 bits of static RAM memory. The board also contains two serial I/O ports; one is intended for "remote" use with a terminal or modem, the other for "local"



use with TI's -301 Microterminal, an EIA terminal, or a teletype. The TM990/101M is available as a -1, -2, or -3. These versions differ in the type of "local" port I/O signals and in the amount of programming of the EPROM memory. The -1, for example, is configured RAM with a TIBUG Monitor Program in the EPROM. All three versions offer three programmable interval timers, up to 17 interrupts and 16 lines of programmable parallel I/O.

Accessories and peripherals for the TM 990 series CPUs include the TM990/301 Microterminal at $125. This terminal interfaces over the EIA series connector to provide hexadecimal entry of program data, as well as display and modification of internal registers and memory under software control. Other accessories include an I/O board, memory expansion boards, a prototyping board and connector kits and cables to interface to selected terminals.

Price for a fully assembled and tested TM990/101M-1 including 1K by 16 EPROM and 1K by 16 static RAM is $625 in single quantities. *Circle No. 140*

Two new software **products for the Tektronix 4051 Graphics Computer System** are available from Leland C. Sheppard. The first is a program development and debugging tool called Documenter-II. According to the company, this product saves time in developing and maintaining 4051 programs. Documenter-II provides formatted listings and several cross-references of 4051 programs on several types of output devices. Prices range from $300.00 for a single copy to $1,500.00 for a facilities license for large installations.

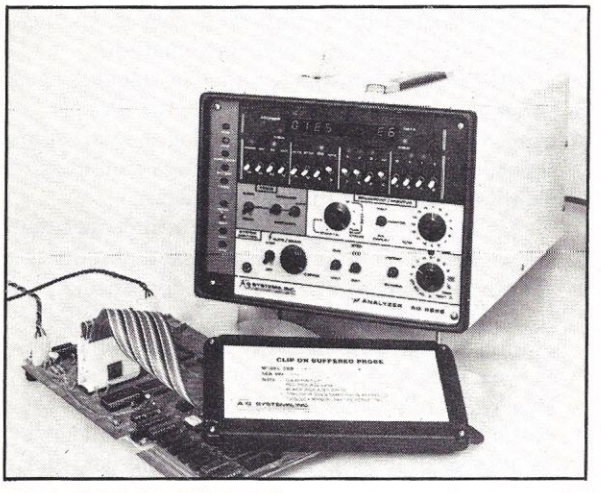
The second product is a PERT charting system called Event Scheduling System (ESS) employing the Critical Path Method (CPM) of scheduling. According to the company, ESS can help organizations

schedule projects and meet deadlines. ESS consists of several programs to build and maintain a project data base and provide charts and reports for the projects. Prices range from $900.00 for a single copy to $3,600.00 for a facilities license for large installations.

Prices for both products include program source and documentation on cartridge tape, user instructions and one year of maintenance. *Circle No. 150*

AQ Systems, Inc., markets the **AQ8080 Microprocessor System Analyzer**, a self-contained, portable instrument compatible with all 8080 system configurations, the company says. A fully buffered 40 pin clip-on probe connects the unit to the microprocessor chip under test. Since the chip is not removed from its own socket, the possibility of chip damage is lessened, and the system is tested in its normal configuration, AQ Systems says.

AQ8080 hardware design does not require memory allocation, address or I/O port assignment, special clock or separate terminal. Built-in controls and displays permit the user to examine or modify all memory locations, I/O ports and internal microprocessor registers, including program counter and stack pointer. Debugging capabilities are provided by conditional breakpoint, data breakpoint and monitor functions that can be qualified to respond to memory reference,



stack reference, I/O locations and read or write activity. Programs may be single stepped by a machine cycle or an instruction step, or run at an adjustable speed of 1 to 4000 steps per second while examining memory or registers. The program trace recorder stores 128 instructions and also provides a block mode of instruction stepping.

The analyzer provides interactive control of the microprocessor and full monitoring of the address bus, data bus and status lines. It can be used for hardware development and test, program debugging, production testing and field service.

Price of the AQ8080, complete with buffered probe, is $2250. *Circle No. 144*

Mind Your Own Business

with computers - personal computers

"Meet my accountant!"

Your business may be manufacturing, engineering or designing panel meters. Or your "business" may be looking for fun ways to improve your golf game – play chess – even make the perfect martini. Whatever your business may be – the computer can be your partner. Let Personal Computing show you how to mind your own business and have fun, too.

Personal Computing

Heath Company has introduced additional **software for the H8 Personal Computer.** Extended Benton Harbor BASIC with file capability, the manufacturer says, is an improvement on the BASIC provided with the H8. It includes character strings, additional convenience and math functions, dynamic storage allocation and access to a real time clock. The new BASIC requires 12K to 18K of memory and comes on audio cassette or paper tape. Specify HC-8-13 (mail order price $20.00) for the cassette and H8-14 (mail order price $10.00) for the paper tape.

New games software for the H8 includes PA-82 Biorhythm, PA-83 Space War and PA-84 Game Set #1, including Craps, Orbit, Tic Tac Toe, Nim, Hexapawn, Hangman, Hmrabi and Derby. Heath says Biorhythm runs under Extended Benton Harbor BASIC and requires 16K of RAM. Space War and Game Set #1 require 24K and 8K of RAM respectively. The games come in cassette form only and sell for $10.00 each. *Circle No. 147*

An **accounting package** for AM and FM radio stations is offered by NCR Corporation.

According to the company, the system helps reduce time and effort in preparing the program log which lists all programs, commercials and other material heard on a radio station. It improves cash flow by automatically preparing customer bills.

The system uses the NCR 499 data processing system employing ledger cards and magnetic tape cassettes for recording and filing information.

After all program and commercial data has been entered on magnetic tape cassettes, the files are merged, creating a daily program log showing all programs, commercials and public-service announcements and the times they are broadcast.

As cash receipts are posted to customer accounts, a separate file on magnetic tape cassette produces an automatic listing for use as a bank deposit slip.

To aid in balancing and auditing, the 499 Radio Station System automatically updates several control accounts when postings are made to customer ledger files. These control ledgers include cash receipts totals, salesman totals, agency commissions paid, special representative commissions paid and financial statement totals.

The software package costs $2400. The NCR 499 system needed to operate the Radio Station System includes a 16K processor, magnetic ledger, two cassette tape handlers and a continuous forms feeder. *Circle No. 148*

Planning to install an air conditioner? HSC Computer Services' **Air Conditioner Selection Program** may help. The program calculates the needed capacity of an air conditioner in BTU/hour, taking into account heat gain through windows, walls, ceiling, floors, electrical equipment and people in the room, as well as heat loss through doors and arches. A correction factor adjusts for location within the United States. *Circle No. 145*

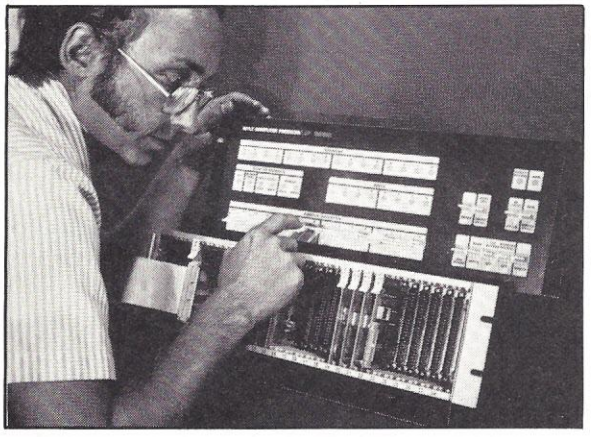
A **floppy disk system** for S-100 bus computers, the Quay 80 Fl system, includes the Q/80 FDC floppy disk controller board (capable of supporting up to four disks), QDOS disk based operating system, the Q/FD1 125 KB 5¼" band-driven disk drive with power regulator and interface cable, and the Q/80 FC floppy disk cabinet. Price is $695 and add-on drives (Q/FD1) cost $350 each. In addition to the floppy disk support, the Q/FDC has available a programmable 8-bit, TTL-compatible, parallel I/O port capable of supporting standard peripheral devices such as line printers, tape punches and keyboards. *Circle No. 138*

Computer Data Systems markets two **disk-based computer systems**, the 3B and an expanded version, the Versatile 4.

This single-unit computer includes a 9" video screen with 24x80 display, built-in mini-floppy disk drive with 143K bytes of storage, upper/lower case alphanumeric keyboard and separate numeric keyboard. The computer mainframe incorporates the 8085 CPU, 24K static RAM and a serial I/O port with RS-232 connector. The Versatile 4 expands this system to 32K static RAM and 315K bytes of storage.

Operating software includes 20K Extended BASIC by Micropolis, a disk operating system and demonstration programs. Five diskettes comprise the software library. The first contains the disk operating system and 20K Extended BASIC. The second diskette contains games including Lunar, Horse Race and Blackjack. Disk #3 contains a small business accounting package which includes programs for financial analysis, mailing list and inventory. Disks #4 and #5 are blank so users can enter their own programs. *Circle No. 141*

Based on the Wyle line of prewired microcomputer rack, drawer and cabinet configurations, Wyle's **microcomputer system configuration guide** allows users to take a building block approach to the design of control and data collection systems. Users can select the required complement of modules for a specific task. Available are serial and parallel digital I/O, analog I/O, communications modules, and over 200 digital logic modules. *Circle No. 151*



CLASSIFIEDS

Wanted: Senior Service Mgr.

Fastest Growing Computer Store on The East Coast. Must be intimate with MIRCO. Software knowledge helpful but not required. Salary to suit. Our staff knows about this ad.

The Computer Factory
790 Madison Avenue
New York, NY 10021
(212) 249-1666

PERIPHERALS, SUPPLIES & SYSTEMS

8" IBM COMP. DISK 3740, S/32	3.95
8", 32 "HARD SECOTR" DISK	3.95
5¼" 1, 10, 16 SECTOR DISK	3.75
COMPLETE EDUCATOR SYSTEM	1000.00
TELETYPE MOD 43, KSR/TTL	900.00
TELETYPE MOD 43, RO/TTL	1350.00
TELETYPE MOD 43, MODEM	784.00
INTERTEC VIDEO TERMINAL	6000.00
COMPLETE SM. BUSINESS SYS.	5600.00

N.Y. residents add sales tax
S&D COMPUTER TECHNOLOGY, P.O. BOX 1283, ALBANY, NY 12201 CALL 24 HOURS/7 DAYS (518) 463-2091

DIRECTORY OF OVER 500 RETAIL COMPUTER STORES

- Fully alphabetized by state and name
- Continuously updated
- Simply copy & cut apart for address labels
- Includes telephone numbers
- Excellent starting point for market researchers manufacturers entrepreneurs

Send $6.00 Check or Money Order to:

Edward Grenda
43 Everett St.
Arlington, MA 02174

MICRO-VERTER

A SPECIAL COLOR MODULATOR FOR APPLE II USERS!!

UHF Version. Operates above channel 14. Eliminates worms!

Operates above the switching harmonics of the computer, thereby yielding a cleaner, worm-free picture. Tunable over a minimum of 4 channels. Interfaces directly with the Apple II as well as most other micros. Comes with video cable and RF output stub coupler. Two-toned cowl type decorator cabinet. Size: 5.5cm x 8.5cm x 11.5cm. Power: +5V. Current approx. 1 ma. Self-powered with 4 pencell batteries. Operating life in excess of 1000 hours or near shelf-life of batteries. Excellent stability. Precise frequency adjustment. No assembly required except for installation of batteries, not supplied. MODEL MVX-500.

AVAILABLE FROM YOUR LOCAL COMPUTER DEALER or direct from ATV RESEARCH. COST including shipping anywhere in USA and Canada — $35.00.

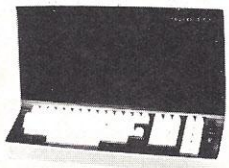
"PIXE-PLEXER" An IC type video-to-RF modulator includes FM sound sub-carrier, color subcarrier and separate R-Y and B-Y inputs. Designed around the LM-1889 chip. A designer's dream with full data sheets. Model PXP-4500. Kit form. $24.50 postpaid.

"PIXE-VERTER" The original computer video-to-RF interface module. Kit form: $8.50 Model PXV-2A

PHONE or WRITE TODAY. DIAL 402-987-3771.

13-PC Broadway **ATV Research** **Dakota City, Nebr. 68731**

Datapoint CRT Terminals



Fully-Assembled — Guaranteed
#3360 $649.50
refurbished

- Add $15.00 Packing
- Shipped F.O.B. Washington, D.C.; Terms Check or M.O.
- Delivery in ten days
- Guaranteed 90 days

Model 3360 speeds from 300-4800 Baud, numeric keypad, Full or half-duplex;

Standard ASCII reed-switch keyboard, with code-able options; green phosphor; 25 80 Ch. lines, addressable cursor; RS-232C serial interface; other speeds available. Manual $10; cable kit $9.95. Datashare/IBM-2260 compatible version $1,100.00.

- Model 3000 $825.00. M-33 ASR Teletype $895, KSR $725; All M-28, 35 components available, also Modems, readers. QUANTITY DISCOUNTS AVAILABLE. Leasing, service at low prices.

TELECOMMUNICATIONS SERVICES CO.
Box 4117, Alexandria, Va. 22303
703-683-4019 / TLX 89-623

WANTED

TRS-80
GAME PROGRAMS

Royalties paid

Send description
Earl J. Bivins
1846 W. Broadway
Anaheim, CA 92804

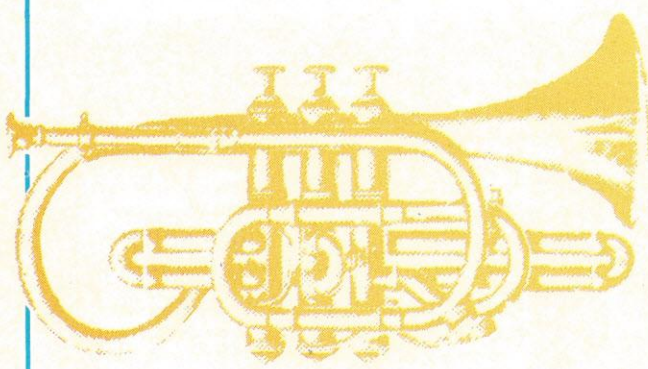
Learn MICROPROCESSOR and MICROCOMPUTER basic send $4.95 per book to

COMPUTER CONCEPTS INC.
PO BOX 641B
HACKENSACK, NJ 07602

FREE – Order now and receive completely worked out 8080 assembly language programs involving – double precision addition and subtraction, multiplication and division subroutines, stack pointer utilization, arrays.

PLUS – 8080 programming aids including hexadecimal machine OP codes, mnemonics and instruction set description.

TRS-80 MONTHLY NEWSLETTER. . . Send for details:
Howard Y. Gosman
Box 149,
New City, New York 10956
New York 10956



Mayor Proclaims "Personal Computing Week" for Chicago Exposition

The '78 Midwest Personal Computing Expo, sponsored by *Personal Computing* magazine, will be held October 5, 6, 7 and 8 in Chicago. In support of the show and in recognition of the impact computers are making in changing American lifestyles, Chicago mayor Michael A. Bilandic proclaimed October 4 - 8 "Personal Computing Week" in the Windy City. Schedule permitting, Mayor Bilandic will also open the show.

Emphasizing practical, cost-cutting innovations for home, small business and professional microcomputer users, the '78 Midwest Personal Computing Expo will feature space-age displays, home entertainment applications and celebrity appearances.

Last year's Midwest Personal Computing Expo drew a crowd of 12,565 doctors, lawyers, business and computer professionals, educators, students, home users and hobbyists.

This year's show, organized by Chicago-based Industrial and Scientific Conference Management, Inc., will be held at the Chicago Loop's Expocenter. With larger exhibition facilities and a promotional budget topping $100,000, show organizers expect more than 20,000 attendees. "And that's a conservative attendance projection," said one show spokesman.

Already, the first stages of the audience promo campaign are underway. A saturation direct mail thrust will reach prime personal computer customers throughout Illinois, Iowa, Minnesota, Wisconsin, Michigan, Ohio, Indiana and other Midwest states.

In addition, since the show's announcement last fall, full page ads have appeared in leading publications, including *Personal Computing, Minicomputer News, Digital Design, Computer Dealer, Chicagoland Business, Electronics Journal, Electronic Packaging and Production* and *Chicago Business Review.*

Three weeks prior to opening, a heavy newspaper ad campaign will break and continue until the final day of the show. And during show week, newspaper show sections will appear in the Chicago dailies and the suburban press. Show promoters are working closely with exhibitor company advertising departments, agencies and PR firms to develop new-product feature stories for area newspapers, business journals and other publications.

TV and radio interview segments will feature exhibitor products, technology advances and details about the exhibition. Billboards, transit advertising, discount coupons distributed by merchants throughout Chicagoland, celebrity appearances and press receptions also are scheduled for the Expo.

Ruann International, with offices in Chicago, Tokyo and London, will direct the total promotional campaign. This advertising and PR agency recently directed promotion for the Chicagoland Business Services and Equipment Expo in April. □

Exhibitors at '78 Midwest PC Expo

Top area and national computing equipment and materials manufacturers and retailers are reserving space for the '78 Midwest Personal Computing Expo in October. Over 40% of exhibit space has already been reserved, and show organizers expect over 150 exhibitors by show time. Exhibitors already signed up include:

Bits N Bytes Computer Stores
Compute-A-Bit
Digital Enterprises
Microcomp
Olson Electronics
Summagraphics
Telpar, Inc.
Artec Electronics
Cherry Electrical Products Corp.
Information Systems, Inc.
Apple Computer Co.
Jade, Inc.
Hobbyworld
Radio Shack
Nabih's, Inc.
Control Data Corp.
Chicago Computer Stores, Inc.
The Milwaukee Computer Store
Chicago Business Review
Quantum Computer Works
Tek-Aids, Inc.
itty bitty Machine Co.
Compucolor Corp.
The Digital Group
Processor Technology
AAA Chicago Computer Center
Computer Electronic Modes
Quay Corp.
People's Computing Co.
Personal Computing
Minicomputer News
Chicagoland Business Services & Equip. News
Electronics Journal
Computer Dealer
Digital Design
Electronic Packaging & Prod.
Newman Computer Co.
Systems & Software
Computer Data Systems
Michigan General Automation
Aspen Computer Co.

ADVERTISERS' INDEX

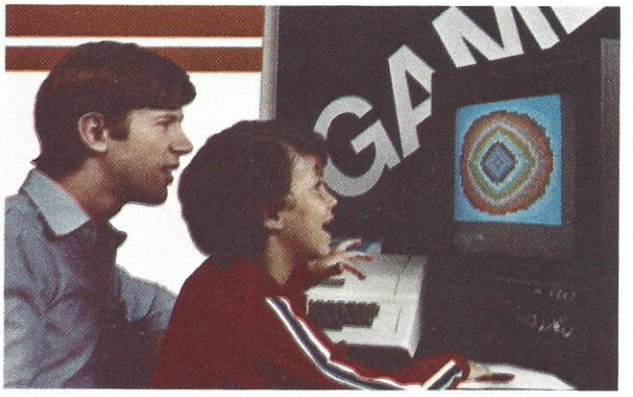
SALES OFFICES

Northeast: George Palken, 1050 Commonwealth Ave., Boston, MA 02215; (617) 232-5470 • **Mid-Atlantic**: Arthur Daks, 104 So. Central Ave., Valley Stream, NY 11580; (516) 872-9550 • **Midwest**: Hank Bean, 2633 Hillside Lane, Evanston, IL 60201; (312) 475-7173 • **Los Angeles**: Carol Stagg, 924 Westwood Blvd., Suite 610, Los Angeles, CA 90024; (213) 478-3017 • **San Francisco**: Ralph Petersen, 1180 Oak Creek Dr., Palo Alto, CA 94304; (415) 328-4392 • **Japan**: Hiro H. Irie, International Business Corp., 11-8, Narita-Higashi 1-chome, Suginami-KU, Tokyo 166; Phone: (03) 311-1746 • Exposition Service Sales Office: Louise Garcia, (505) 898-4562.

Rated G

The Best Game in Town.

Welcome to ComputerLand. An incredible adventure into the world of personal computers. A one-of-a-kind shopping experience.



Each ComputerLand store presents everything you ever wanted to know about computers. And then some.

Take our Game Room, for starters. You'll find excitement for the whole family in our endless variety of challenging computer games. You can battle the Klingons in an out-of-this-world game of Star Trek. Create an electronic work of art with a computer controlled TV. Test your skill in a game of computerized hangman.

You can even plot your biorhythm.

But we're more than just fun and games. Each ComputerLand store offers a knowledgeable and personable staff of professionals

to serve you.

Plus the greatest available selection of micro components. Whether it's a data processing system for your business or a computer controlled sprinkler system for your home, you'll find whatever you need at ComputerLand.

Read on.

Genuine Service.

We want to supply you with the one system that's right. Rather than a complete system that isn't. Or a limited system that is.

That's why, at ComputerLand, you deal with real professionals who are also real people. People who speak your language . . . in addition to BASIC, COBOL or FORTRAN.

People, in short, who can offer both the novice and the old hand the same expert guidance in selecting the optimum system he or she needs.

Yet, assisting in the purchase is only the beginning of ComputerLand's service. If the kit you bought requires a little more do-it-yourself than you yourself can do, we provide assembly assistance.

If that complex program proves to be just that, we provide programming assistance.

And if your system breaks down, our in-store service department will get you back up and running.

Right now!

Great Selection.

Your first stop at ComputerLand may well be your last stop.

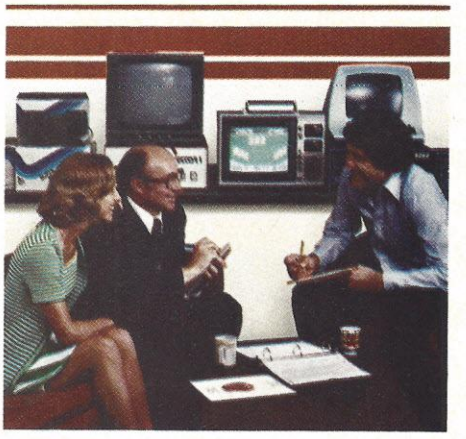
ComputerLand offers the finest quality and largest selection of all the major brand names. Like Apple Computer, Cromemco, DEC, Diablo, Hazeltine, ICOM, IMSAI, Lear Siegler, National Semiconductor, North Star, Texas Instruments, Vector Graphics and more.

Plus a complete inventory of tools, books and accessories.

What's more, at ComputerLand, we deal in product. Not promises. Our inventory is on our own shelves. Rather than the manufacturer's. So you can take delivery on tomorrow's components today.

Which means, simply put, that at ComputerLand, you get exactly what you want.

Exactly when you want it.



Be Our Guest.

Begin with the grand tour of our exhibit areas. "Test-drive" any of our individual systems.

Then tell us your needs. We'll sit down and talk about the system that's right for you. It's as easy as that at ComputerLand.

The great computer store. RATED G.

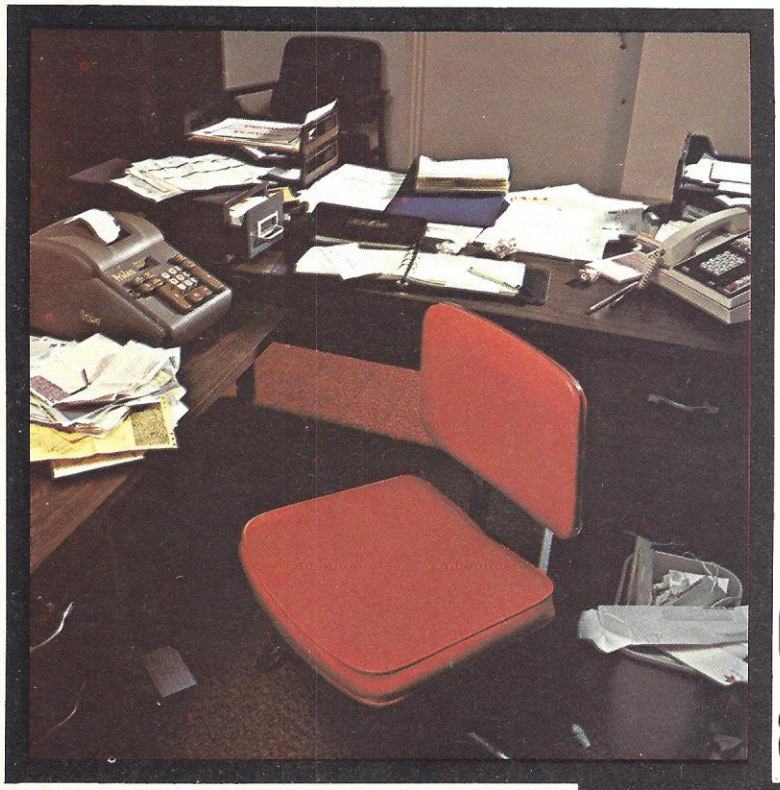
Call or write for the address of the ComputerLand store nearest you. Franchise opportunities available.

ComputerLand™

14400 Catalina St., San Leandro, CA 94577 (415) 895-9363

CIRCLE 2

Is your TRASH FLOW higher than your CASH FLOW?



SOFTWARE LIBRARY

Five diskettes are included to give you immediate operating and programming capabilities.

DISK 1 is a master diskette with BASIC, MDOS, Text Editor, Assembler and more!

DISK 2 has many games including LUNAR, CRAPS, and SPLAT. There's room left for you to add your own.

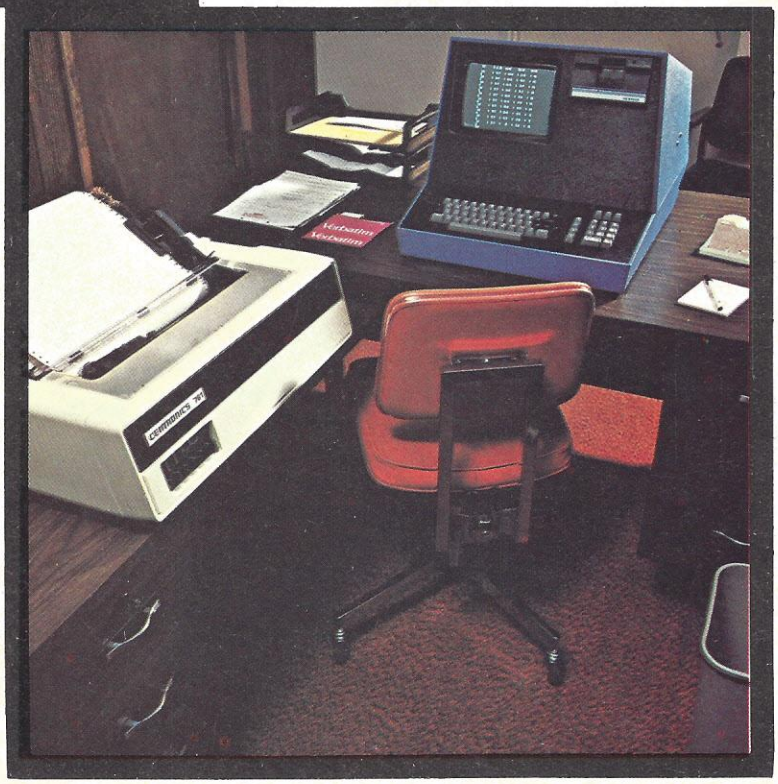
DISK 3 contains a Small Business Accounting package which includes Accounts Payable and Receivable, Inventory, a General Ledger, and more.

DISKS 4 & 5 are blank so you may add your own programs.

Clean it up with a VERSATILE 3B™

If you're a businessman, we know how difficult it is to keep neat and efficient accounting records. Let our **VERSATILE** systems do it for you. You'll have a complete system built into a single cabinet, and a free software library on diskette to get your computer working for you the first day. $3295 Assembled and tested.

WHAT ELSE DOES A COMPLETE SYSTEM NEED?



Contact Our Distributors for Regional Sales and Service

Alexander & Co., Inc.
5518 Florin Road
Sacramento, CA 95823
(916) 422-9070

Huron Electronics
415 N. Silver
Bad Axe, Mich. 48413
(517) 269-9267

The Computer Store
3801 Kirby Dr., Suite 432
Houston, Texas 77098
(713) 522-7845

Southeast Representative:

Scientific Sales Co.
175 W. Wieuca, Suite 210
Atlanta, GA 30342
(404) 252-6808

DealerApplications Available.

COMPUTER DATA SYSTEMS

5460 Fairmont Drive • Wilmington, Delaware 19808 • 302-738-0933

CIRCLE 3